DTIC FILE COPY

1

AD-A217 938

JMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a	1b. RESTRICTIVE MARKINGS NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFIT/CI/CIA-89-005
6a. NAME OF PERFORMING ORGANIZATION AFIT STUDENT AT North Carolina State Univ	6b. OFFICE SYMBOL *(If applicable)*
7a. NAME OF MONITORING ORGANIZATION AFIT/CIA	
6c. ADDRESS *(City, State, and ZIP Code)*	7b. ADDRESS *(City, State, and ZIP Code)* Wright-Patterson AFB OH 45433-6583
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL *(If applicable)*
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO | WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)* (UNCLASSIFIED)
Simulation and Animation of the Cherry Point, NC, Naval Aviation Depot Proposed Engine Blade and Vane Rework Facility

12. PERSONAL AUTHOR(S)
Waring L. Worsham, Jr.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
THESIS/~~DISSERTATION~~	FROM ________ TO ______	1988	99

16. SUPPLEMENTARY NOTATION
APPROVED FOR PUBLIC RELEASE IAW AFR 190-1
ERNEST A. HAYGOOD, 1st Lt, USAF
Executive Officer, Civilian Institution Programs

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

DTIC
S ELECTE D
FEB 12 1990
D

90 02 12 039

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE *(Include Area Code)*	22c. OFFICE SYMBOL
ERNEST A. HAYGOOD, 1st Lt, USAF	(513) 255-2259	AFIT/CI

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

AFIT/CI "OVERPRINT"

Simulation & Animation

of the

Cherry Point, NC

Naval Aviation Depot

Proposed Engine Blade & Vane

Rework Facility

Waring L. Worsham, Jr.

November 10, 1988

North Carolina State University

Accesion For

NTIS CRA&I ☒
DTIC TAB □
Unannounced □
Justification ....

By ....
Distribution /

Availability Codes

Dist | Avail and / or Special

A-1

DTIC
COPY
INSPECTED

Contents

Abstract

Title: Simulation and Animation of the Cherry Point, NC, Naval Aviation Depot Proposed Engine Blade and Vane Rework Facility

Author: Waring L. Worsham, Jr., Captain, USAF

Completed: November 10, 1988

Degree Awarded: Master of Science, Industrial Engineering
North Carolina State University

Number of Pages:

30	Pages in Main Report
29	Pages in a User/Technical Reference
39	Pages in 3 Attachments
1	Page Bibliography

99	Total Pages

The Naval Aviation Depot at Cherry Point, NC, repairs damaged aircraft engine blades. Plans to build a new engine blade repair facility capable of processing roughly 350 blades per hour are presently being reviewed. We wanted in this effort to create an interactive atmosphere that allows simple use of several powerful tools capable of providing important insight to those responsible for the many decisions that need to be made in such an undertaking

A linear programming model had previously been created and is now being used to provide deterministic modeling of the present facility. An interactive data base editor used in conjunction with the linear program leads the user through the creation of the data files required to run the LP models.

Building upon the linear programming model, we created a stochastic model using the SIMAN simulation software package. SIMAN was chosen because CINEMA, its animation companion, allows SIMAN models to be "brought to life" on a personal computer screen. ~~To date,~~ in addition to the linear program, we have created a simulation model and an animation of the proposed facility.

To tie everything together and allow for ease of use, we wrote two additional programs. The first, written in C, automatically creates the data files necessary to perform the simulation by drawing from the LP data bases. The second, written in BASIC, is an interactive menuing program that calls the specific application (and data files when appropriate) the user selects in response to menu prompts. This report provides the details of these efforts.

Cherry Point Naval Aviation Depot Simulation/Animation Project

Introduction

The Naval Aviation Depot at Cherry Point, NC, repairs damaged aircraft engine blades. Presently, plans are in the making to build a new engine blade repair facility capable of processing approximately 350 blades per hour. Our goal was to create an interactive atmosphere that allows simple use of several powerful tools capable of providing important insight to those responsible for the many decisions that need to be made in such an undertaking.

A linear programming model had already been created and is presently being used to provide deterministic modeling of the present facility. An interactive data base editor in conjunction with the linear program leads the user through the creation of the data files required to run the LP models. The linear program is explained in detail in its own separate documentation.

We wanted in this effort to build upon the linear programming model and create a stochastic model using the SIMAN simulation package. SIMAN was chosen because CINEMA, its companion animation package, allows SIMAN models to be "brought to life" on a computer screen. To date, in addition to the linear program, the simulation model, and an animation of the facility have been accomplished. The particulars of the simulation and animation are provided in later sections.

The first thing that needed to be done was to get a feel for how much data would be produced and/or used by all of the above. It was desired that all packages draw from the same data base of information if possible. The data estimate included within this report indicated that the amount of data anticipated was not significantly large enough to require at this point the design any special data base. What was presently being created and used by the Linear Programming model and its data editor would be adequate provided some minor changes were made to take into consideration additional data required by the simulations not necessary for the LP. These changes are currently being accomplished under a separate effort.

Having accomplished the above it became desirable to make the creation of simulations and animations as simple as possible to the end user. A program has been written for the automatic creation of the files necesarry to perform the simulation. At present it takes the necessary information from the files created by the LP data bases and translates them into the files necessary to run SIMAN. That information necessary for the simulations that is not present in the LP data bases for now is drawn from data files created separately.

The only thing that remained at this point was to create an easy-to-use interactive menu that allowed the end user to perform all of this from a single environment. A menu program has been written that calls the program (and data files when appropriate) the user selects in response to menu prompts. The next page provides a summary of the project objectives.

Project Objectives

1. Estimate Data Requirements

   a. Design a Data Base if Necessary

2. Modify the Linear Program Programs

   a. Data Design
   b. Upper and Lower Bounds on the LP

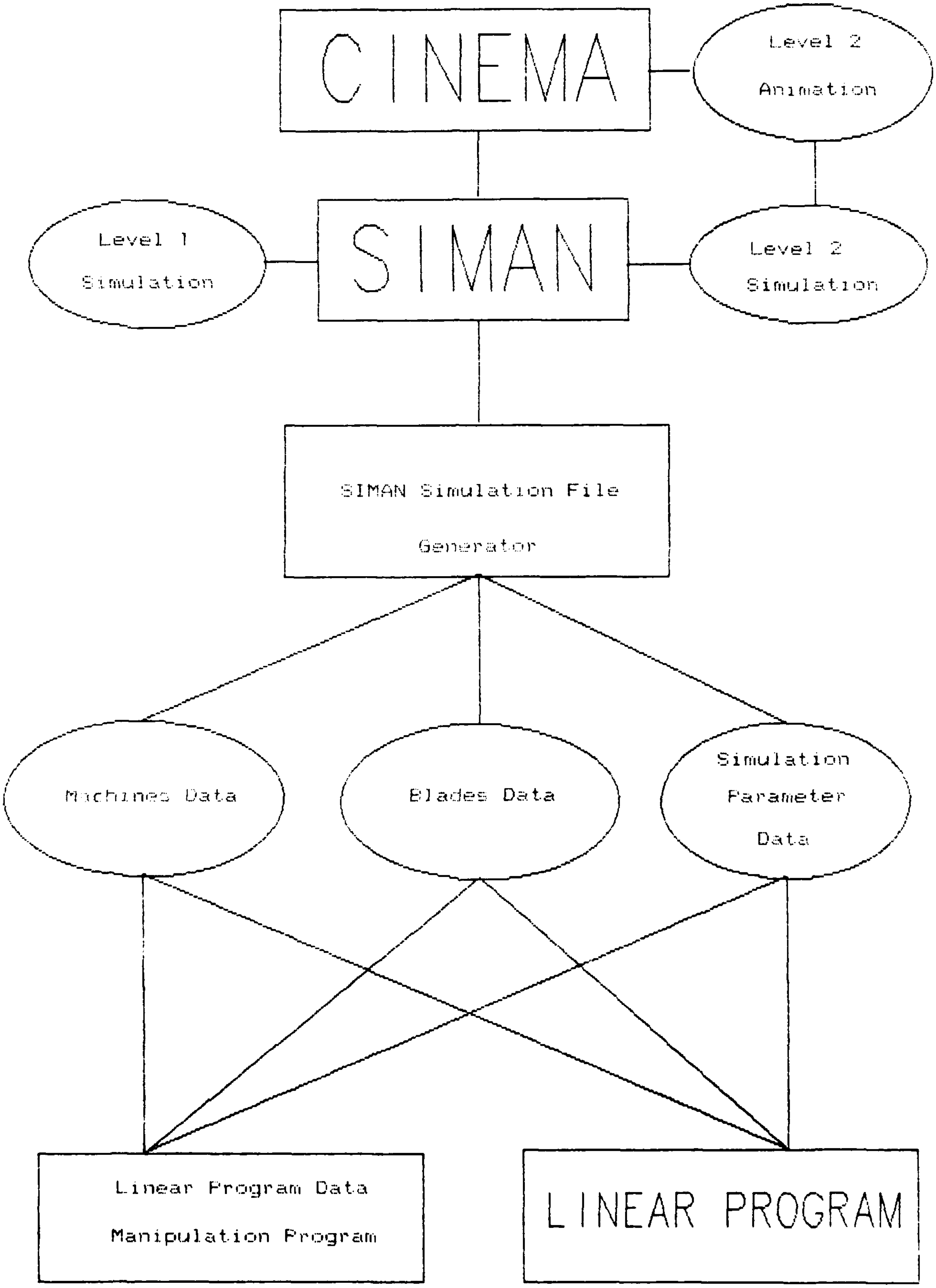
3. Build Simulations

4. Automate Simulation Data File Generation

5. Animate the Simulations

6. Build an Interactive Menu File to Run Everything

7. Produce Necessary Documentation


CINEMA
Level 2 Animation
Level 1 Simulation
SIMAN
Level 2 Simulation
SIMAN Simulation File Generator
Machines Data
Blades Data
Simulation Parameter Data
Linear Program Data Manipulation Program
LINEAR PROGRAM

The Linear Programming Model

The Linear Programming Model of the Naval Aviation Depot blade and vane rework facility provides a deterministic solution to the product mix problem associated with processing the various batches of blade types on the various available resources While it is not my intent to go into the details of how this program works and what information it provides, I would like to discuss it briefly in the context of how it was built upon in this project. It is the predecessor effort and portions of what was accomplished in the development of the LP was carried over into the development of the simulation.

The linear program provides an "optimal solution" to problem of determining the optimal mix of blades by placing a priority (value) on each blade type and maximizing the return realized by multiplying the priority of each blade type by the total number of blades of the respective type processed. Symbolically:

$$\text{Objective Function:} \quad \text{Max } Z = \sum_{i=1}^{n} Ci * xi$$

$$\begin{array}{lccccccc}
\text{Subject To:} & a_{11}x_1 & + & a_{12}x_2 & + & \text{-----------} & a_{1n}x_n & \leq b_1 \\
 & a_{21}x_1 & + & a_{22}x_2 & + & \text{-----------} & a_{2n}x_n & \leq b_2 \\
 & \vdots & & \vdots & & & \vdots & \\
 & a_{m1}x_1 & + & a_{m2}x_2 & + & \text{-----------} & a_{mn}x_n & \leq b_m
\end{array}$$

Ci = Priority (Value) of Processing One of Blade Type i

xi = Number of Type i Blades Processed

aij = Amount of Resource i Required to Process One Blade Type j

bi = Capacity of Resource i

This model provides a means of determining the optimal blade mix to process based on the relative measure of merit of producing each blade type. It also provides a quick look at whether or not resource capacities can possibly meet demand -- a minimum requirements look based purely on capacity available vs. capacity needed. It does not, however, give insight into the dynamics of the system -- how the system changes over time and what impact these changes have on the overall outcome. Nor does it take into consideration the effects of variability (i.e., breakdowns, etc.) The simulation was therefore written and the animation created to provide this insight.

Overview of SIMAN

SIMAN uses two types of files, the "Model" and the "Experiment", that are written and compiled separately and later linked together to perform a particular simulation.

The Model File:

The MODEL file is the file that contains the simulation logic. The commands used in this file control flow of entities through the system being modelled. The Model provides the simulation framework; it describes the basic simulation scenerio such as entity creation, activity, and deletion. The idea here is that regardless of how often entites are created, how fast they are processed, and how often they are destroyed, the simple fact remains that they must be created, processed, and destroyed. The commands used in the Model file are called BLOCKS. Model files are identifed by default with a ".MOD" extension in their file name. This file can be created with any text editor that can save in ASCII. Once created, the Model file must be compiled using SIMAN's model compiler (syntax: "MODEL filename.MOD"). The compiled Model file is given a ".M" extension.

The Experiment File:

The Experiment file on the other hand, with few exceptions, contains the numeric portion of the simulation. The information contained in the Experiment file, as its name implies, can be looked upon as the experimental portion of the simulation, providing the grounds for "what if" types of questions. It holds the data describing such specifics as where, how many, how long, and how often. Parameters called for in the model file are stored in the Experiment file. The Experiment file also sets simulation limits (i.e., maximum number of queues and entities and simulation run length) and defines output contents. The commands used in the experiment files are called ELEMENTS. Experiment files are identified with a ".EXP" extension in their file name. This file can be created with any ASCII text editor. Once created, the Experiment file must be compiled using SIMAN's experiment compiler (syntax: "EXPMT filename.EXP"). The compiled Ex,eriment file is given a ".E" extension.

The Program File:

Once the Model file and Experiment file have been written and compiled they must be linked together. SIMAN's linker program (syntax: "LINKER filename.M filename.E filename.P") accomplishes this. Filename.M and filename.E are the compiled Model and Experiment files respectively and filename.P is the default name for the linked Program file created by the linker program. (It is convenient (and our practice) to use the same filename for all of the various files created for each individual simulation -- distinguished only by appropriate extension.) This

Program file is very important; it is this file that can now be used with SIMAN and CINEMA to perform simulation and animation

Running the Simulation:

It is the Program file that contains the information that the SIMAN processor needs to perform simulation. The command "SIMAN filename.p" causes SIMAN to run the simulation using the information in the Program file. By default, output is printed only to the screen. To make SIMAN save results to an output file for later review, simply use the DOS convention for redirection of output: (syntax: "SIMAN filename.p > filename.out").

The GOSIMAN Batch File:

For convenience, we have created a batch file to compile both the Model and Experiment files, link them together, run SIMAN on the resultant program file, and direct the simulation output to an output file. The batch file requires that the operator provide a single filename parameter. It assumes that all files for a given simulation have a common filename and use the extension conventions outlined above. The proper syntax is "GOSIMAN filename". After creating the Output file, GOSIMAN deletes the compiled Model and Experiment files since they are no longer needed (any changes to the Model and Experiment files would require re-compilation anyway). The program file is kept for possible future use in animation. Below is a summary of the files created/used by SIMAN.

The GOSIMAN Batch File

```
   Syntax:  GOSIMAN filename
  ------------------------------
    c:\siman\model %1.mod
    c:\siman\expmt %1.exp
    c:\siman\linker %1 %1
    c:\siman\siman %1.p > %1.out
    del *.m
    del *.e
    type %1.out
```

SIMAN Simulation Files Summary

Convention	File Description
filename.mod	The Simulation Model File
filename.exp	The Simulation Experiment File
filename.m	The Compiled Simulation Model File
filename.e	The Compiled Simulation Experiment File
filename.p	The Simulation Program File
filename.out	The Simulation Output File

The NADEP Simulation

We have approached the NADEP simulation in a stepwise manner. We have created three scenerios called for simplicity level 1, level 2, and level 3. Although philosophically the same facility is being modelled, each level is distinguished by its own set of assumptions. Some assumptions were common to all levels:

a. Scheduling is uniform based on yearly induction quantities. This type of scheduling is not a requirement of the simulation by any means. It provides a computationally convenient estimate and can be changed if deemed appropriate.

b. Batch arrivals, machine and agv breakdowns (when modelled), machine and agv repair (when modelled), and batch processing times are all exponentially distributed. This assumption also can be changed in the future if deemed appropriate.

c. Blade batches are processed first in/first out (FIFO). Again, alternative processing acenerios can be used if desired.

I will now describe each of these levels in more detail in the following sections.

Level 1

Level 1 is the simplest version of the simulation. It was designed to provide information for the planning and specification of the engine blade rework facility. It also represents the control simulation, used to verify Level 2 changes as they were made. It provided an initial, high-level look at machine utilizations and turnaround times.

The following additional assumptions apply to the level 1 simulation:

1. Instantaneous transfer of materials (i.e., no transportation system used to transfer batches from one workcenter to the next). When a batch finishes being processed at one workcenter it immediately begins processing at the next workcenter.

2. Blade batches follow fixed routes.

3. No breakdowns.

4. No attrition. All blades that start processing complete processing. There are no losses.

5. Infinite queue sizes. There is no limit to the number of batches that can be waiting for a particular kind of resource.

The level 1 simulation is accomplished using the following scenerio.

At Beginning

Create batches of blade types. Mark time of creation
Assign batch type and batch size to attributes.
Assign appropriate route sequence to the batch.
Send batch to first station in route sequence.

At Stations

Mark batch arrival time to station.
Place batch in station queue.
Seize first available machine.
Tally batch wait time in station queue.
Process the batch.
Release the machine.
Go to next station in route sequence.

At End

Record batch total time in system.
Remove batch from simulation.

Level 2

Level 2 is the more sophistocated version of the above scenerio. It provides a more refined look at machine requirements, includes a transportation system, identifies and drops out irreparable blades (attrition), and allows for machine and transportation system breakdowns.

Level 2 built gradually upon the Level 1 assumptions to include each of the following assumptions:

1. A transportation system is included. Batches of blades are transported between stations by the AGV closest to the batch upon completion of processing. Other selection rules are available.

2. Attrition is included. Batches drop out irreparable blades.

3. Machine and AGV reliability is included; breakdowns and repairs are modelled using exponential distributions. Repairs are performed by a capacitated pool of repairmen.

4. Capacitated queueing is a goal but as yet has not been accomplished. SIMAN provides no simple procedure for blocking of full queues. Batches arriving to full queues would have to be balked elsewhere or destroyed.

The level 2 simulation is accomplished using the following scenerio:

At Beginning

Create batches of blade types. Mark time of creation.
Assign batch size to an attribute.
Assign appropriate route sequence to batch.
Start batch at entrance station.

At Entrance Station

Increment counter for number of batches in system.
Queue batches for transport to stations.
Request agv for transport to first station.
Transport to first station in sequence when agv arrives.

At Machine Stations

Mark arrival time of batch.
Free the agv that transported the batch to the station.
Place batch in station input queue.
Seize a machine resource when available.
Record wait time in machine (input) queue.
Process the batch.
Decrement batch size to account for attrition.
Release the machine.
Mark machining completion time.
Place batch in station agv (output) queue.
Request agv for transport to next station.
Record wait time in agv queue when batch seizes agv.
Transport to next station in sequence.

At Exit Station

Free the agv that transported the batch to the exit station.
Record individual batch times in system.
Decrement counter for number of batches in system.
Record batch summary times in system.

At Machine Breakdown Decision Blocks

Create entity to cause station breakdowns.
If all machines are down, dispose of the breakdown entity.
Otherwise, Go to Machine Breakdown/Repair Block.

Machine Breakdown/Repair Block

Decrement Number of Up Machines by 1.
Place machine in repair queue. Mark arrival to queue.
Seize a repairman when available.
Record wait time in repair queue.
Repair the machine.
Free the repairman when repair is complete.
Increment # of machines by 1.

At AGV Breakdown Decision Blocks

Create an agv breakdown entity.
If agv is already down, dispose of breakdown entity.
Otherwise, Go to AGV Breakdown/Repair Block.

At AGV Breakdown/Repair Block

Halt agv.
Repair the agv.
Re-activate the AGV after repair.

Level 3

Although levels 1 and 2 were the focus of this project, it is forseen that an "operational simulation", level 3, will ultimately be accomplished. This level of simulation will provide information of the more day to day nature, useful once the facility has actually been designed, built, and become operational. The following set of goals define level 3:

1. Daily (or weekly) run.

2. Initial conditions (startup) accounted for.

3. Variable routing.

4. User-defined scheduling of batches through the system.

The Simulation Output

The power of the simulation is realized in the simulation output. Once the simulation has been created and verified to be a realistic model of the system being analyzed, the output can be analyzed to determine what effect various decisions and changes have on the overall system. Although we were not provided with any concrete data to conduct and analyze a particular study (our goal, in fact, was not to actually analyze the system for NADEP, but to create a means for them to do so themselves), certain points can be made about just what the simulation output provides.

The information supplied in the simulation output allows the user to analyze the value or cost of various decisions and alternatives. The "what if" questions can be answered by inputing the appropriate changes into the data base (which in turn updates the SIMAN experiment file), re-running the simulation, and comparing the output to previous simulations run under different conditions. If the new simulation appears to provide significantly different results, a re-animation may be appropriate to provide a visual representation of what effect the change makes. In many instances, however, the effect of the change can be seen very simply by analyzing the simulation output.

The output provided for the NADEP blade and vane rework facility can be broadly classified into two categories: tallies and discrete change variables. Tallies are more typically the programmer-defined statistics created by the programmer in the model file while discrete change variables are typically those tracked automatically (though possibly manipulated by) the programmer. The output for both provides essentially the same type of statistical information (average value, standard deviation, minimum value, maximum value) for the variables defined. Let's now look at how this information can be used.

The following pages provide tables describing each of the types of output displayed in the simulation output. The information provided by these statistics can be used as a basis for rational decisions concerning such things as machine and transportation system requirements, batch flow times, queue sizes, and system throughput. Although the list of statistics provided is fairly comprehensive it is by no means exhaustive. The output can be tailored, once the end user defines more specifically what interests him, to present most any type of pertinent statistic desired as needs change and interests dictate.
A few examples will illustrate the type of information that can be obtained by reviewing the simulation output.

A high amount of wait time for a machine might indicate eitner that the demand for that resource is higher than can be reasonably expected or that the machine reliability is poor and needs to be improved. At any rate, a potential bottleneck has been identified for whatever reason. To determine which of the mentioned possibilities is more likely the case, inspect the "number of machines up" statistic for the station representing that resource. If 6.95 of the 7 available machines on average are not broken down, it is very probable that machine reliability is not the primary cause of the problem (although certainly a minor contributor). A very high utilization of the resource should be anticipated and can be verified easily by inspection of the appropriate machine utilization statistic. (Be careful to recognize the SIMAN utilization statistics are expressed in terms of the number out of the total of available machines that are being used, not the percentage.) Given the present methods, either additional capacity needs to be added, another shift or a subcontractor considered, or perhaps alternative routings and processes away from the bottleneck area should be used.

The associated statistics for agv queues, up time, and utilization provid essentially the same type of information as it applies to the transportation system. The evaluator should be careful to realize that an increased number of transporters does not necessarily translate into increased throughput. Consider five o'clock rush hour traffic. The output provides information about how the presently modeled situation performs; it does not make any evaluations of possible solutions.

The "wait for repair" statistics provides information about maintenance resources. High wait times and large repair queues could indicate either a scarce repair resource or a lot of machines with reliabilities that need to be improved, if possible.

The flowtime statistics provide the important insight into turnaround times for the various types of blades being processed through the facility. Flowtimes significantly higher than the total amount of processing time might indicate that the blades are being held up at and that there is a potential bottleneck Perhaps the transportation system is not efficient. Possibly, batches are lining up for an unreliable resource. The statistics listed above can be checked to verify whether or not any of these are actually the case.

Finally, the "number of batches in system" and the "number of batches through system" provide exactly what they say they provide. If the number of batches in the system consistently grows with longer and longer simulation runs, there is obviously some kind of capacity problem; batches are arriving to the system faster than they are leaving the system. This is a serious problem and machine and agv statistics should be studied to determine where the problem(s) might be located. The "number of batches through the system" statistic gives an indication of whether or not desired throughput is being or can be attained. Care should be taken when interpreting this statistic; understand that initial conditions (i.e., an empty system) can cause the statistic to be artificially low for short simulation runs since it takes some time for an empty system to ramp up to any significant level of work.

Tally	Description
WAIT FOR MACH X	The amount of time a batch has to wait in a queue for available machine resource type X.
WAIT FOR AGV @ X	The amount of time a batch has to wait in a station's output queue (after processing by resource X) for an agv to transport the batch to the next station.
TYPE X FLOWTIME	The amount of time required for a batch of blade type X to be completely processed through the system.
OVERALL FLOWTIME	The amount of time required for batches of blades in general to be completely processed through the system.
WAIT FOR REPAIR	The amount of time a machine must wait when down for an available repairman.

Discrete Change Variable	Description
# IN STN QUE X	The number of batches waiting for a resource type X in the station's input queue.
UTIL OF STN X	The number of resource type X in use at any given time.
# IN AGV QUE X	The number of batches waiting for an agv in the output queue (after processing) of station X
# AGVS UP	The number of agv's up (not broken down) at any given time.
# BUSY AGVS	The number of agv's in use over time (up and not idle).
# OF MACH @X UP	The number of machines that are up (not broken down) over time.
# IN REP QUEUE	The number of machines broken down waiting for repair at any given time (broken down, waiting for an available repairman.)
UTIL OF REPMEN	The number of reparimen busy at any given time.
BATCHES IN SYS	The number of batches physically in the system at any given point in time.
BATCHES THRU SYS	The total number of batches processed through the system over the course of the simulation run. (Maximum Value statistic of interest here.)
BLADES THRU SYS	Same as Batches Thru Sys except the batches through the system are multiplied by their attrited, final batch size.

Overview of CINEMA

CINEMA is the program used to create and edit animation files. Cinema makes use of five different animation files: the "Layout", the "Resource", the "Entity", the "Transporter", and the "Palette". The Layout file is the resultant merger of all of the five animation files and is used in conjunction with the SIMAN Program file to perform a particular animation using the CSIMAN program.

The Layout File:

The LAYOUT file is the file containing the animation "floorplan". This file, in conjunction with the SIMAN program file, provides all the information needed by CSIMAN to perform a particular animation. The animation is defined in the Layout file by the resources, entities, transporters, and the (color) palette added to the layout during a Cinema session. The layout may also include text/titles, lines defining machining center/aisles, and timers/scales to track simulation progress. Layout files are identified by default with a ".LAY" extension.

The Resource File:

The RESOURCE file is the file containing the pictures of the resources available for use during an animation. These resource drawings correspond to the resources used in the simulation. The drawings must be created using the Cinema resource editor. It should be emphasized that the Resource file, once created, is simply a storage place for resource drawings. In order to use these resources during an animation they must first be placed on the layout during a Cinema session. If changes are later made to a resource, the old resource must be removed from the layout and the new, changed resource must be re-placed. Resource files are identified by default with a ".RES" extension.

The Entity File:

The ENTITY file is the file containing the pictures of the entities available for use during an animation. An entity is any object being created by and moved through a simulation. The entity drawings must be created using the Cinema resource editor. It should be emphasized that the Entity file, once created, is simply a storage place for entity drawings. In order to use these entities during an animation they must first be added to (though not necessarily placed on) the layout. If changes are later made to an entity, the old entity must be removed from the layout and the new, changed entity must be added. Entity files are identified by default with a ".ENT" extension.

The Transporter File:

The TRANSPORTER file is the file containing the pictures of the transporters available for use during an animation. These transporter drawings correspond to the transporters used in the simulation. The transporters must be created using the Cinema transporter editor. It should again be emphasized that the Transporter file, once created, is simply a storage place for transporter drawings. In order to use these transporters during an animation they must first be added to (though not necessarily placed on) the layout. If changes are later made to a transporter, the old transporter must be removed from the layout and the new, changed transporter must be added. Transporter files are identified by default with a ".TRA" extension.

The Palette File:

The PALETTE file is the file containing the sixteen color palette available for use during an animation. The Palette file may be edited using the Cinema palette editor. If no palette is defined, the default palette is used (DEFAULT.PAL). As with the other files, the Palette file is simply a storage place for the color palette. In order for an edited palette to be used during an animation it must first be added to the layout. If changes are made to the color palette, it must be re-added to the layout. Palette files are identified by default with a ".PAL" extension.

The GOCINEMA Batch File:

A Cinema editing session is started by typing "CINEMA" at a DOS prompt in the directory containing the "CINEMA.EXE" file. Obviously, if the directory containing "CINEMA.EXE" is in the current DOS path, you can execute Cinema from any DOS prompt. Once Cinema has started the individual files to be loaded and edited. The files can alternatively be loaded upon execution of Cinema by providing the filenames as parameters (syntax: "CINEMA filename.lay filename.res filename.ent filename.tra filename.pal"). We have created a batch file to simplify this. The GOCINEMA batch file accepts one parameter, "filename". The proper syntax is "GOCINEMA filename". It assumes that all files for a given layout use the default Cinema extension conventions outlined below.

The GOCINEMA Batch File

```
            Syntax: "GOCINEMA filename"

  cinema %1.lay %1.res %1.ent %1.tra %1.pal
```

CINEMA Animation Files Summary

Convention	File Description
filename.lay	The Animation Layout File
filename.res	The Animation Resource File
filename.tra	The Animation Transporter File
filename.ent	The Animation Entity File
filename.pal	The Animation Color Palette File

Important Note:

We have had no success in having Cinema load all five animation files during startup using the Cinema procedure for accomplishing this. Once started, Cinema consistently fails to recognize the resource and palette files and prompts the user to re-enter them. Since our batch file attempts to take advantage of this procedure it suffers the same weakness. When the names are re-entered during program startup at the prompt, however, Cinema loads them readily. It appears to be simply a bug in the program. Cinema spokespersons were not aware of the problem before we called and asked about it. It does not affect the performance of Cinema or its animations but does provide an inconvenience to the user since he is requred to manually ensure that the resource and palette files are loaded. For now, simply typing the filename (including directory path) with the appropriate extension when prompted appears to be the only solution. Loading three files out of five automatically is still better than having to load all five manually, so for now we continue to use this less than perfect procedure.

The GOCSIMAN Batch File:

GOCSIMAN assumes that the above processes have already been accomplished. It also assumes that an appropriate layout file has been defined and created using CINEMA. It also assumes that an appropriate program file has been produced by SIMAN. Using CINEMA's layout file (filename.lay) and SIMAN's program file (filename.p), GOCSIMAN loads the layout and program files, displays the layout on the screen, and provides a menu of options to the operator prompting him for inputs to begin the animation. (The appropriate syntax without using the GOCSIMAN batch file is: "CSIMAN filename.lay filename.p")

The GOCSIMAN Batch File

```
Syntax: "GOCSIMAN filename"

 csiman %1.lay %1.p
```

CSIMAN Files Summary

Convention	File Description
filename.lay	The Animation Layout File
filename.p	The Simulation Program File

The NADEP Animation

The animation of the blade and vane rework facility, upon completion of the simulation models, was simply a matter of creating the arena for activities taking place in the simulations to be displayed. The CINEMA layout file was created in the manner described in the Overview of CINEMA. What the layout ultimately looks like depends simply and ultimately upon the skill and creativity of the animator.

The Layout

We attempted to make the layout represent as closely as possible the most recent layout proposed for the blade rework facility. Where discrepancies occurred between machinery placed on the drawings and the machine capacities listed in the LP data files, the drawings were used and the simulations adjusted to account for the differences.

Since only seven blade types are presently being modelled using fourteen different types of machine resources, only those machines and entities required for the animation were placed on the layout (All anticipated resources are drawn in the level 2 resource file, but only the fourteen needed for the present animation were actually placed on the layout -- see "Overview of SIMAN"). In so doing, only one screen was needed for the necessary resources, greatly simplifying the design and visualization of the animation. The overall plant view produced allows the big picture perspective of the facility's activities. A sample of the animation layout, indicating among other things the placement of resources within the facility, is provided at the end of this section. Also provided are sample sketches of the entities (batches) and transporters (AGV's).

Because the layout contains all of the necessary machining centers for blade types one through seven, not much room remained for anything else once the resources had been placed. Queues are therefore represented by scales indicating how full the queues are at any particular time during the animation instead of by a line of entities forming outside the needed resource. This does provide a benefit of keeping the screen from being cluttered with the many individual boxes representing blade batches. Queue levels assume capacities of forty batches as indicated in the linear programming data files even though the simulation presently allows infinite queue sizes (SIMAN provides no simple blocking procedure for full queues). A level approximately half full indicates a current queue size of twenty.

Stations were placed on the layout close to the resources associated with them. Stations are used to define distances between pickup and dropoff points. CINEMA requires that every distance be defined by drawing a path on the screen identifying the route the transporter is to take in going from one station to the next. It uses the distance from the appropriate DISTANCES element and the agv speed from the TRANSPORTERS element in the

SIMAN experiment file to calculate the speed at which the agv travels across the screen. Although SIMAN assumes that the distance from station A to station B is the same as from station B to station A, CINEMA requires that both be drawn on the layout. The path from A to B must be defined, and the path from B to A, although the same, must be defined (in reverse order).

We take advantage of color in the animation to help show the status of machines and agv's. Each machine resource and each agv can be in one of three states at any given time: idle, busy, or down. If the resource or agv is idle but operational, it is gray; if busy, green; if broken down, red. As an agv or machine changes status, its color changes to indicate the change.

Finally, a simulation clock is placed in the top center of the screen to aid in realizing passage of time. One complete revolution of the dial represent a single eight-hour workday.

Some Finer Points of Animation

Producing the animation not only provides a nice visualization of the simulation process, but probably more importantly, the opportunity for the simulation programmer to convince himself that what he thinks is happening in the simulation is actually happening. This visual verification process proves extremely valuable and perhaps in itself justifies the effort. Several simulation logic problems were identified by the animation that might otherwise have been difficult to detect or might otherwise have gone unnoticed. Results displayed on the screen and in the simulation output can be accepted with greater confidence than might otherwise have been realized.

There are some problems, however, with producing an animation of the rework facility nature. To begin with, the range of times required for the various activities varies widely. For example, the time it takes for an AGV travelling at a moderate speed (something less than an average walking pace) to travel from one station to another is going to be significantly less than the amount of time required to run a batch of blades through a cleaning/plating line. The former may require approximately two minutes, while the latter may require two days. This poses a problem with time scaling the animation.

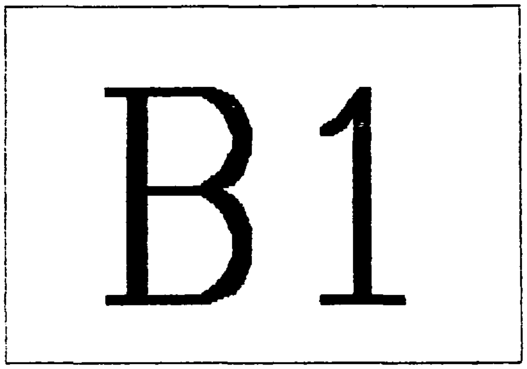
For example, if the timing of the simulation were scaled such that the time for the cleaning/plating process took two minutes instead of two days (a reduction to 0.2% its original time), a similar reduction in agv transport time would have the agv traveling across the screen in approximately 1/4 second. On the other hand, if the simulation were sped up only to the point of allowing visualization of agv transport from station to station (say, five seconds instead of 120), the cleaning/plating process would require forty minutes. While this may not seem too bad from the outset, consider the possibility of waiting forty minutes to see one agv travel across the screen for five seconds. To visualize a single day of animation would require 20 minutes; a single week nearly two hours.

Because of the above situation, it became necesarry to slow the agv to a level significantly lower than their actual speed in order to actually see them as they travel across the screen. Although this does not provide a realistic visualization of the process (it may take as long as three days for an agv to travel from one station to another), it does allow the person viewing the animation to verify that the agv's are being called properly according to the agv selection rule (in this model, closest available agv is called) and that the batches are being moved to the appropriate next station in its route sequence.

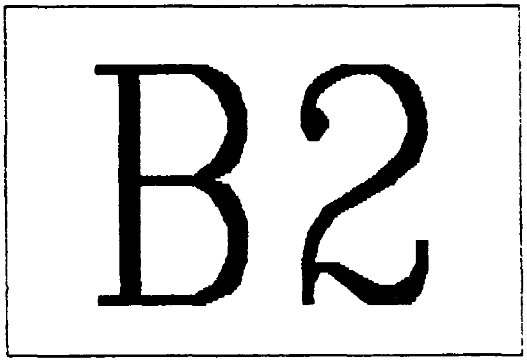
It should be emphasized that the proper way of animating a simulation is to first have in mind during simulation creation that an animation will also be created and that what might be the best way to accomplish the simulation is possibly not best for the animation. Where simplifying assumptions might be perfectly acceptable during simulations, the same assumptions might prove inappropriate for animations.

For example, a simulation might simply assume an individual grouping of common resources constitutes a "station". While the manner in which entities travel from one station to the next is of no real consequence to the simulation, only that they get from one station to the next in the estimated amount of time required for transport, it is of great consequence to the animation. Entities are transported across the screen from station to station and the exact placement of these stations on the layout determine the look of the animation. It may not be appropriate to assign an individual station to each group of resources; it may be appropriate to assign two. In the cases of agv's it may even be appropriate to define segments of tracks as resources to be seized by agv's so that the possibility of collision is avoided. Unfortunately, the stepwise manner in which resources became available for the simulation and then the animation and the way in which the two are typically learned does not lend itself to such an approach for beginners.

Entity Symbols

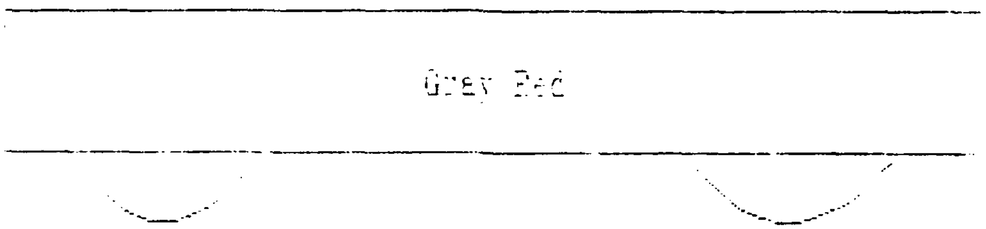


Batch Type 1

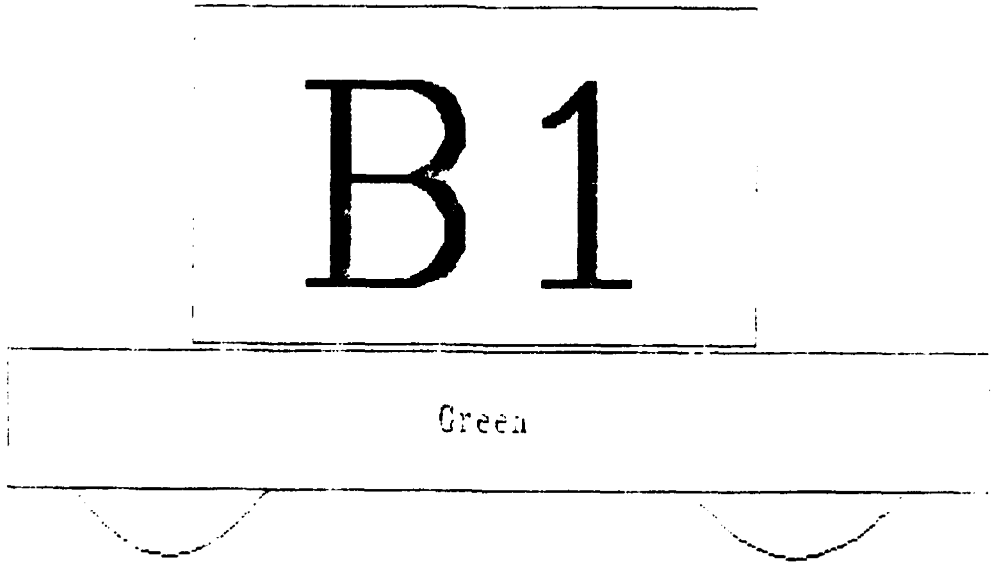


Batch Type 2

Transporter (AGV) Symbols



Idle/Inactive



Busy

The Cherry Point Naval Aviation Depot Menu

User Reference

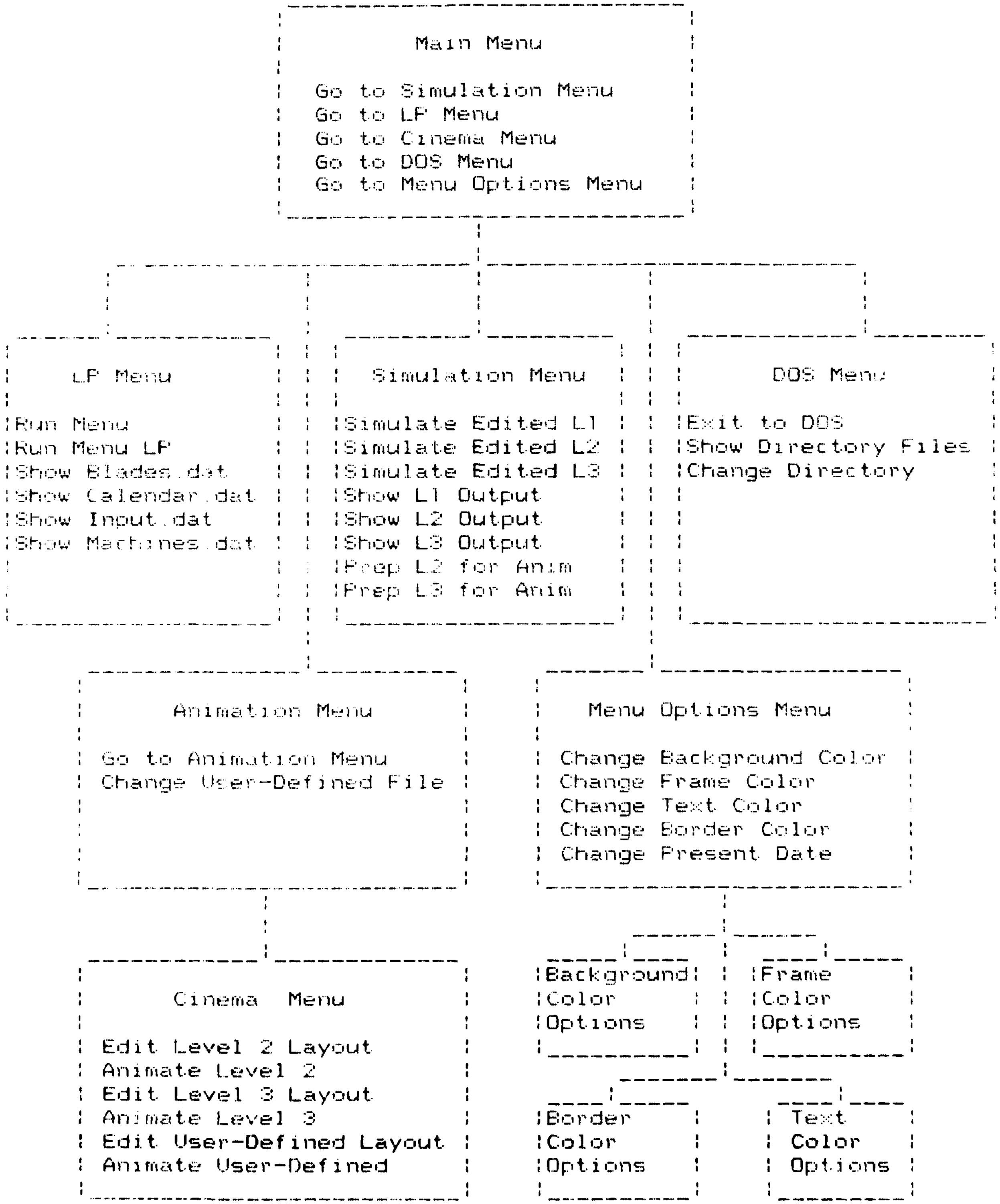
The Cherry Point Naval Aviation Depot Menu provides easy access to all of the actions necessary for altering, recompiling, and running SIMAN simulations and CINEMA animations. It also provides menu color options and selected DOS capabilities, including the ability to review the listing of files in the current directory and to exit temporarily to DOS while keeping the menu in RAM. The current directory and the current date are always displayed with the menu. Both can be changed by the user if desired or necessary.

Loading and Using the Menu

The menu is loaded by executing the "GOMENU.BAT" batch file. Use of the menu is as simple as pressing the keyboard key identified for the desired option. The menu uses a tree structure, guiding the user through the menu options. A keystroke will either begin an action, cause a beep if there is no option associated with the keystroke, or take the user to another menu of options. At any level other than the Main Menu level, pressing escape or <Q> will return the user to the previous menu (see note below). Pressing escape or <Q> at the Main Menu causes the menu program to be terminated and returns the user to the root directory and the DOS prompt. At this point, the menu is removed from memory and must be restarted by executing the "GOMENU" batch file. If the menu directory is placed in the autoexec.bat path statement the menu can be loaded from any DOS prompt. Since the menu is used in conjunction with SIMAN and CINEMA, it is designed to be saved to and used with the hard disk necessary to run the simulations and animations. It assumes that the menu files are stored in a directory named "NADMENU" and that all other directories are named as indicated at the end of the menu technical reference.

Note: Keyboard keys are identified within inequality signs. For example, <Q> represents the keyboard key Q.

Menu Tree



Main Menu

Go to Simulation Menu
Go to LP Menu
Go to Cinema Menu
Go to DOS Menu
Go to Menu Options Menu

LP Menu

Run Menu
Run Menu LP
Show Blades.dat
Show Calendar.dat
Show Input.dat
Show Machines.dat

Simulation Menu

Simulate Edited L1
Simulate Edited L2
Simulate Edited L3
Show L1 Output
Show L2 Output
Show L3 Output
Prep L2 for Anim
Prep L3 for Anim

DOS Menu

Exit to DOS
Show Directory Files
Change Directory

Animation Menu

Go to Animation Menu
Change User-Defined File

Menu Options Menu

Change Background Color
Change Frame Color
Change Text Color
Change Border Color
Change Present Date

Cinema Menu

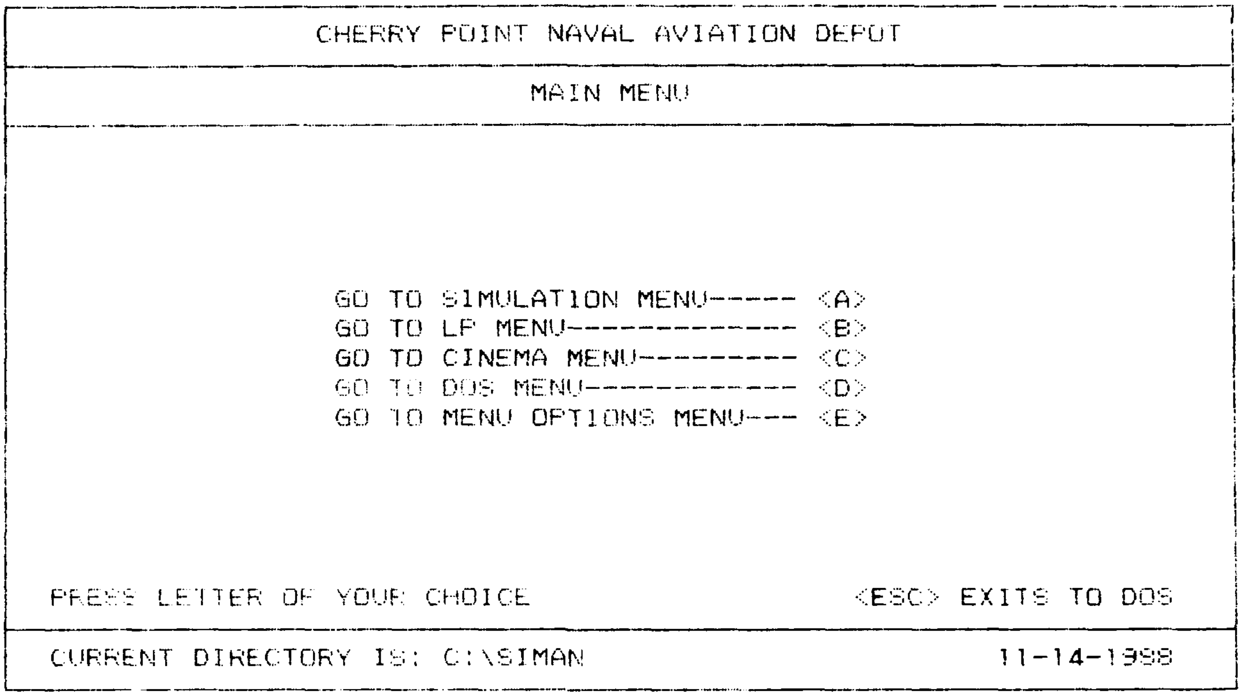
Edit Level 2 Layout
Animate Level 2
Edit Level 3 Layout
Animate Level 3
Edit User-Defined Layout
Animate User-Defined

Background Color Options

Frame Color Options

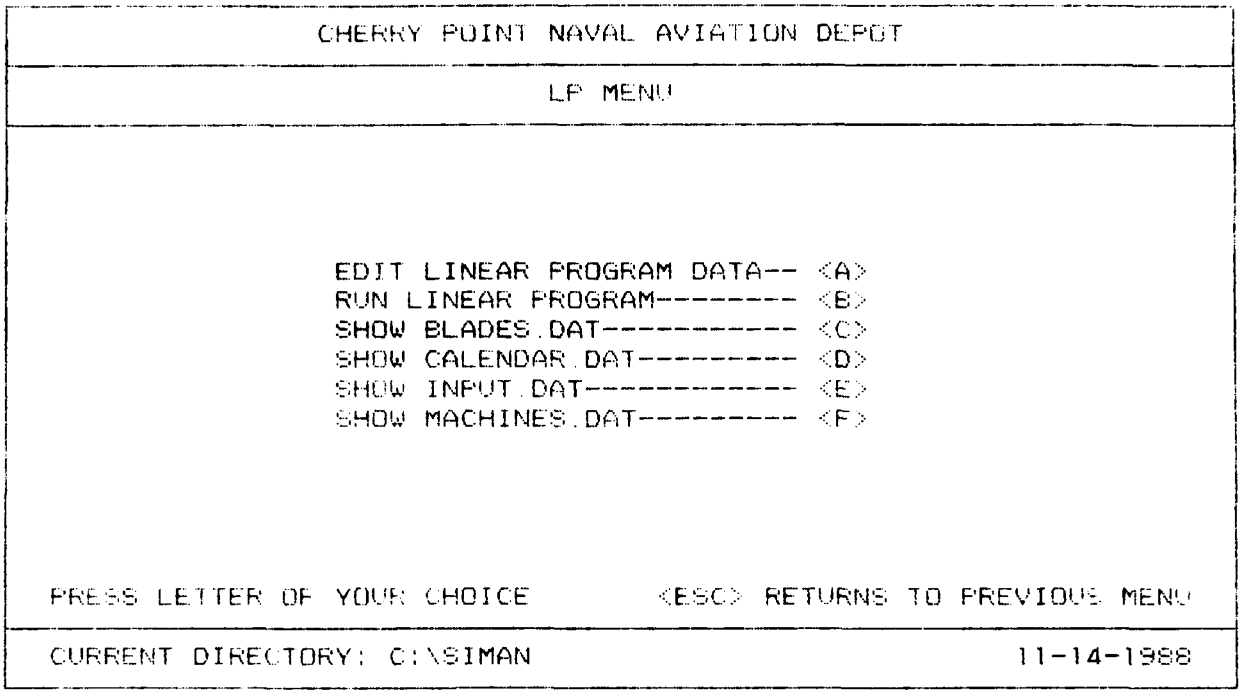
Border Color Options

Text Color Options



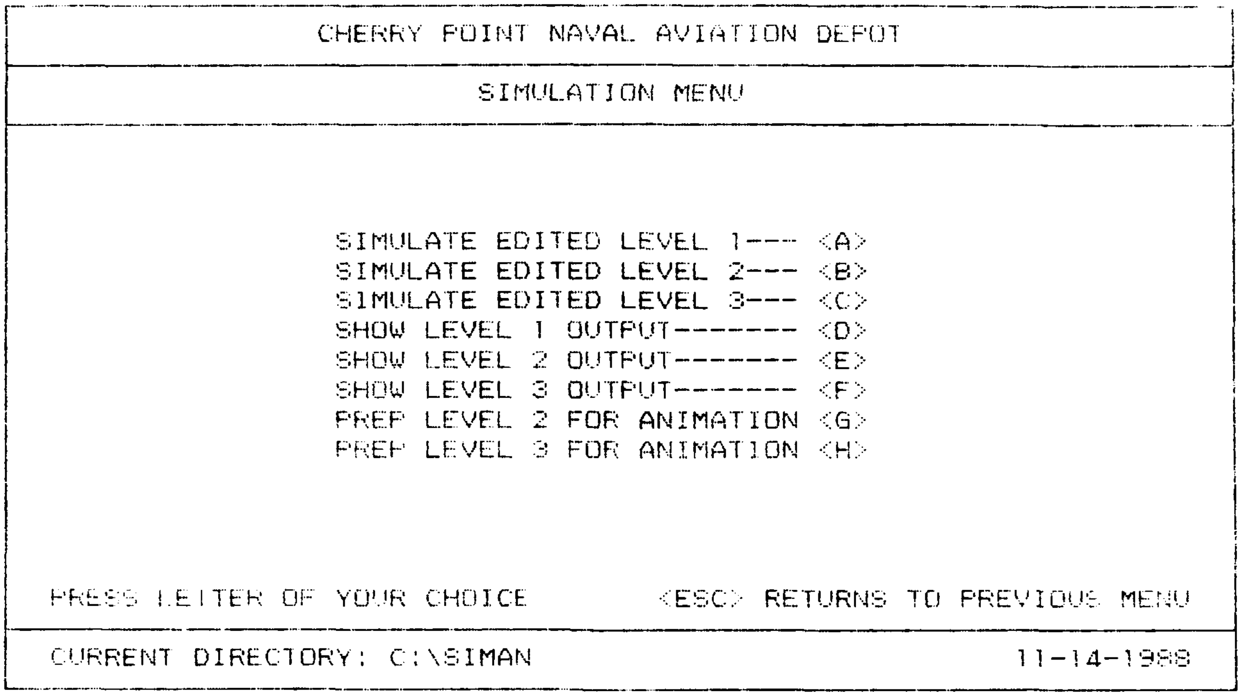
```
                    CHERRY POINT NAVAL AVIATION DEPOT

                                MAIN MENU



                GO TO SIMULATION MENU------ <A>
                GO TO LP MENU-------------- <B>
                GO TO CINEMA MENU---------- <C>
                GO TO DOS MENU------------- <D>
                GO TO MENU OPTIONS MENU---- <E>



  PRESS LETTER OF YOUR CHOICE                     <ESC> EXITS TO DOS

  CURRENT DIRECTORY IS: C:\SIMAN                          11-14-1988
```

The Main Menu is the initial menu the user sees upon start up of the NADEP menu program. Each of these listed options takes the user to a second-tier menu, discussed in more detail on the following pages. Pressing Escape at this menu terminates the menu program and returns the user back to the root directory DOS prompt.



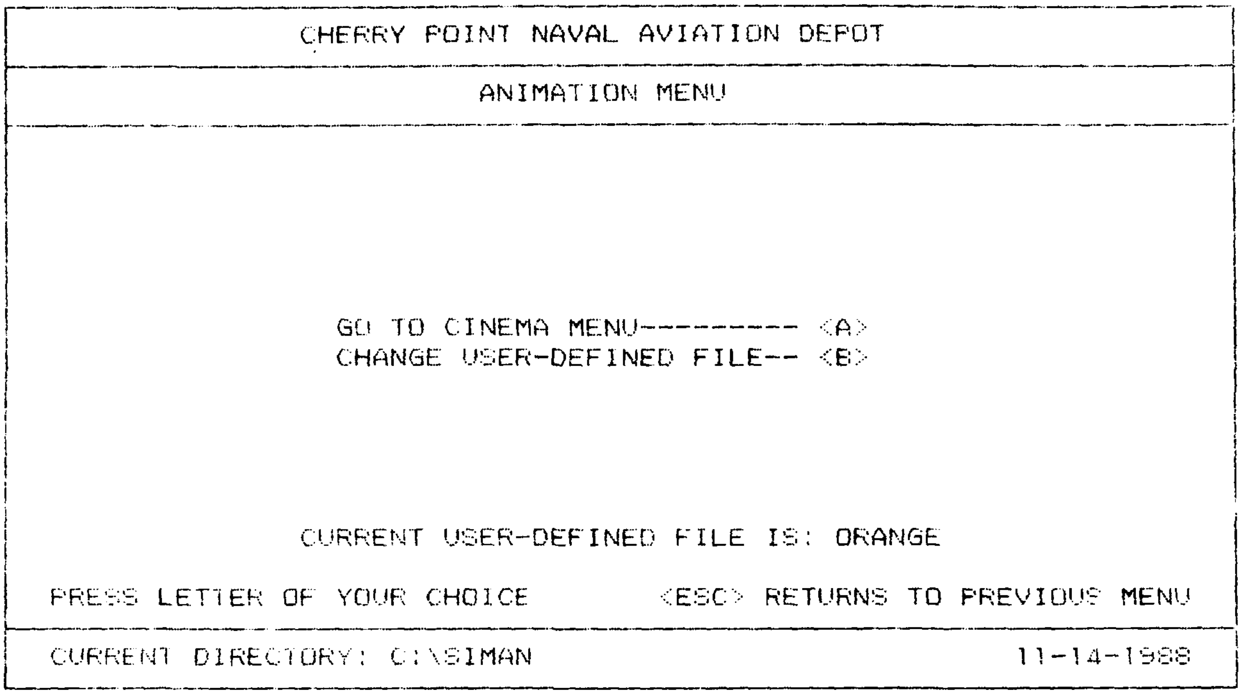
```
                    CHERRY POINT NAVAL AVIATION DEPOT

                                LP MENU



               EDIT LINEAR PROGRAM DATA-- <A>
               RUN LINEAR PROGRAM-------- <B>
               SHOW BLADES.DAT----------- <C>
               SHOW CALENDAR.DAT--------- <D>
               SHOW INPUT.DAT------------ <E>
               SHOW MACHINES.DAT--------- <F>




  PRESS LETTER OF YOUR CHOICE        <ESC> RETURNS TO PREVIOUS MENU

  CURRENT DIRECTORY: C:\SIMAN                                11-14-1988
```

The Linear Program Menu provides via option <A> access to the LP "MENU" program. This option allows the user to edit the data files used by the LP and simulation program runs. Option <B> allows the user to actually run the edited Linear Program. Options <C> through <F> allow the user to view the present contents of the data files prior to entering the editing or linear programming functions. Escape returns the user to the Main Menu.



```
                    CHERRY POINT NAVAL AVIATION DEPOT

                             SIMULATION MENU



                   SIMULATE EDITED LEVEL 1--- <A>
                   SIMULATE EDITED LEVEL 2--- <B>
                   SIMULATE EDITED LEVEL 3--- <C>
                   SHOW LEVEL 1 OUTPUT------- <D>
                   SHOW LEVEL 2 OUTPUT------- <E>
                   SHOW LEVEL 3 OUTPUT------- <F>
                   PREP LEVEL 2 FOR ANIMATION <G>
                   PREP LEVEL 3 FOR ANIMATION <H>



  PRESS LETTER OF YOUR CHOICE        <ESC> RETURNS TO PREVIOUS MENU

  CURRENT DIRECTORY: C:\SIMAN                                11-14-1988
```

Options <A> ,<B>, and <C> are used to run the indicated simulation This would be necessary if changes had been made to the LP data and the impact of the changes on the simulations were of interest. The GOSIMAN batch file which reads the data from the LP data files is called, creates the Model and Experiment files necessary to run the simulation, and runs the simulation which creates the SIMAN program and output files. The new program file may then be used to create animations Before it can be used, however, it must first be copied into the cinema animation subdirectory. Options <G> and <H> accomplish this. The output files contain the results of the simulation run and can be reviewd by selecting option <D>, <E>, or <F>. Escape returns the user to the Main Menu.

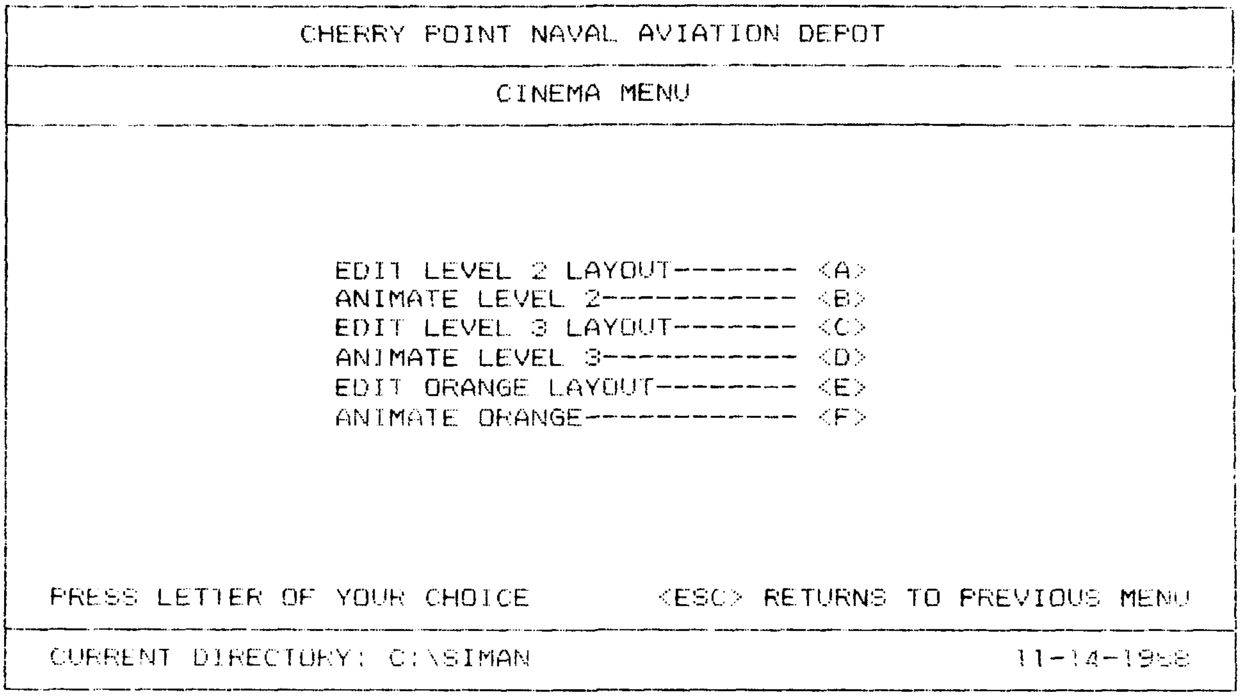


```
                     CHERRY POINT NAVAL AVIATION DEPOT

                              ANIMATION MENU




                      GO TO CINEMA MENU--------- <A>
                      CHANGE USER-DEFINED FILE-- <B>




                   CURRENT USER-DEFINED FILE IS: ORANGE

  PRESS LETTER OF YOUR CHOICE        <ESC> RETURNS TO PREVIOUS MENU

  CURRENT DIRECTORY: C:\SIMAN                                11-14-1988
```

This menu provides an intermediate step before proceding to the Cinema Menu. It allows the user to select something other than the Level 2 and Level 3 Blade/Vane Rework Facility animations. If option <B> is selected, the user is prompted to enter the layout and program filenames of the user-defined animation and simulation files. The user need not adhere to the common naming convention using option <B>. For example, it is possible to animate a level 1 simulation on a level 3 layout. (Obviously, however, the program file must be a valid compiled and linked simulation program file with extension "p" and the layout file must be a valid Cinema animation layout file with extension "lay".) Once the filename has been entered and confirmed the user has the option of proceeding to the Cinema Menu by choosing option <A> or returning to the Main Menu by pressing Escape.

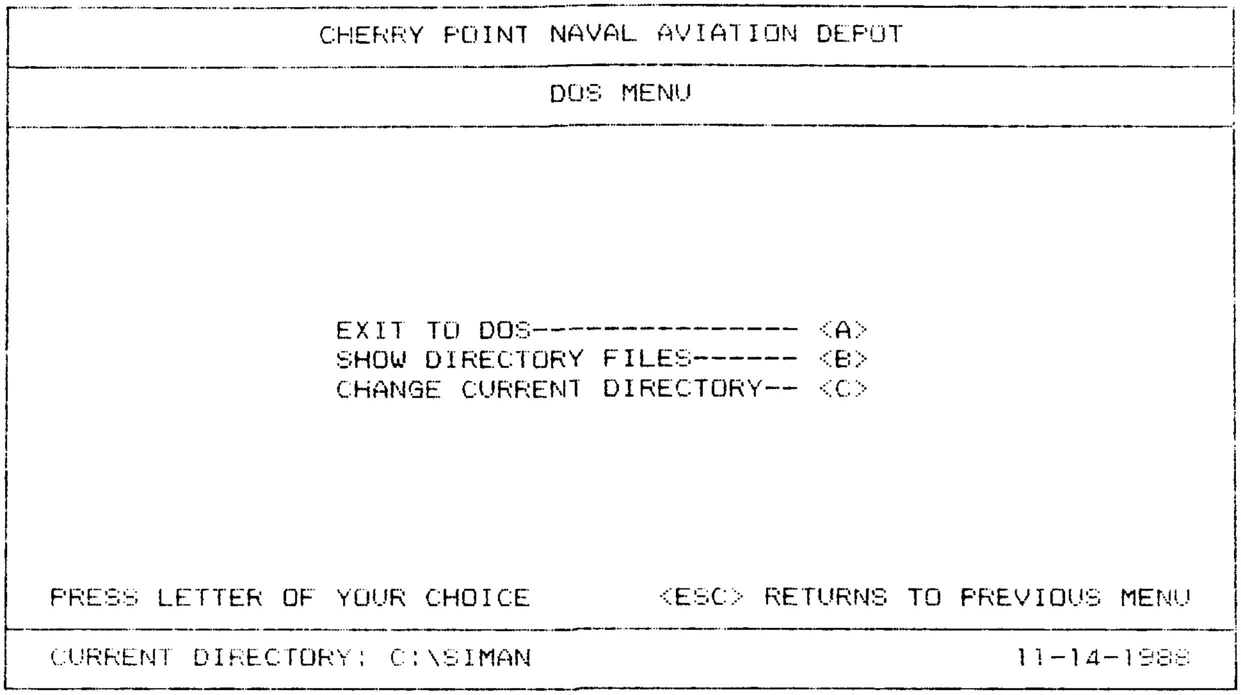
Important Note:

If option <B> is selected, the user must answer the filename prompts by providing a filename (without extension). If the user decides he does not wish to define an alternative animation, simply press return at the filename prompts and answer "Y" to the confirmation. At this point no alternative file is selected and the user is returned to the animation menu.



```
                CHERRY POINT NAVAL AVIATION DEPOT

                           CINEMA MENU



                EDIT LEVEL 2 LAYOUT------- <A>
                ANIMATE LEVEL 2----------- <B>
                EDIT LEVEL 3 LAYOUT------- <C>
                ANIMATE LEVEL 3----------- <D>
                EDIT ORANGE LAYOUT-------- <E>
                ANIMATE ORANGE------------ <F>



 PRESS LETTER OF YOUR CHOICE        <ESC> RETURNS TO PREVIOUS MENU

 CURRENT DIRECTORY: C:\SIMAN                              11-14-1988
```

Cinema is the animation program used in conjunction with the SIMAN models (see Overviews of SIMAN and CINEMA). Cinema provides the means for creating and editing the animation layout. CSiman takes the input layout created during a Cinema session and the program file created by SIMAN and actually produces the animation on the screen. Options <A>, <C>, and <E> allow the user to edit either the level 1 layout, the level 2 layout, or the predefined (from the Animation Menu) user layout These options call Cinema and load the appropriate files for editing. Options <B>, <D>, and <F> call CSiman and load the appropriate Siman Program and Cinema Layout files.

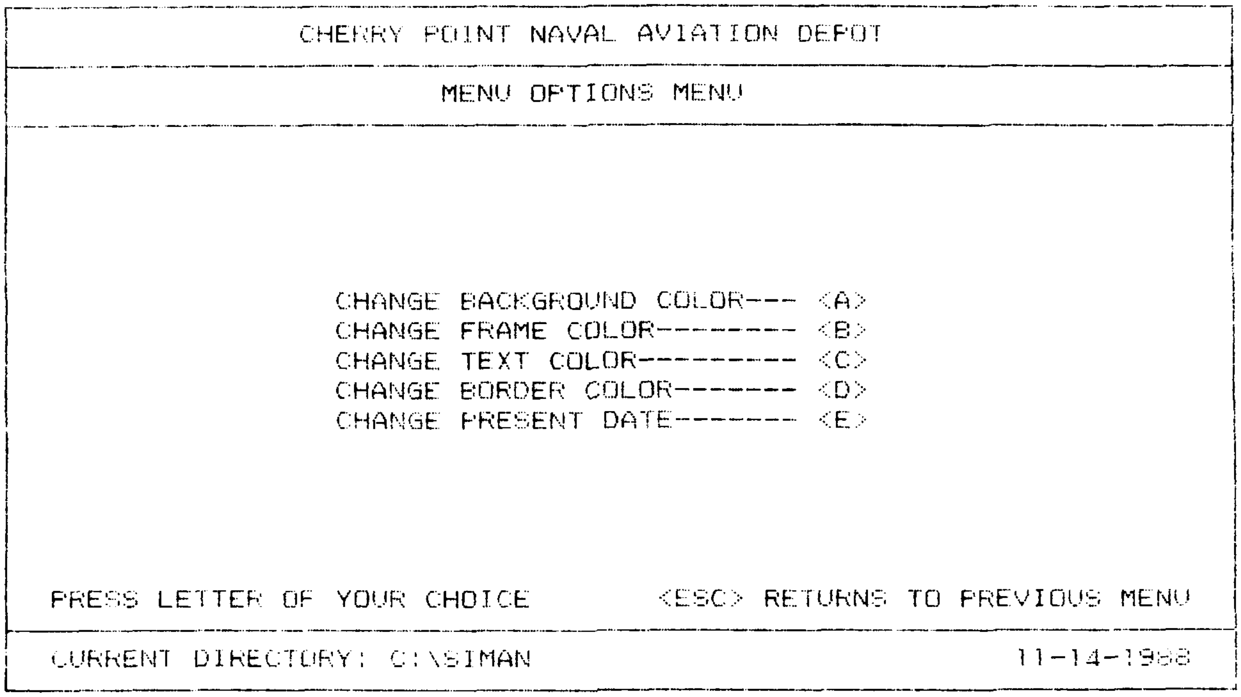


```
                    CHERRY POINT NAVAL AVIATION DEPOT

                                DOS MENU




                  EXIT TO DOS--------------- <A>
                  SHOW DIRECTORY FILES------ <B>
                  CHANGE CURRENT DIRECTORY-- <C>




  PRESS LETTER OF YOUR CHOICE       <ESC> RETURNS TO PREVIOUS MENU

  CURRENT DIRECTORY: C:\SIMAN                               11-14-1988
```

The above menu allows the user to change the current directory and view the files in that directory without leaving the menu. Should it be necessary to perform more extensive DOS functions, the menu provides a DOS gateway, a means of temporarily exiting the menu program. Once in DOS using option <A> the user can perform whatever DOS commands necessary and can return to the menu by simply typing "EXIT" at any DOS prompt. The menu remains in memory during this course of action and need not be reloaded.

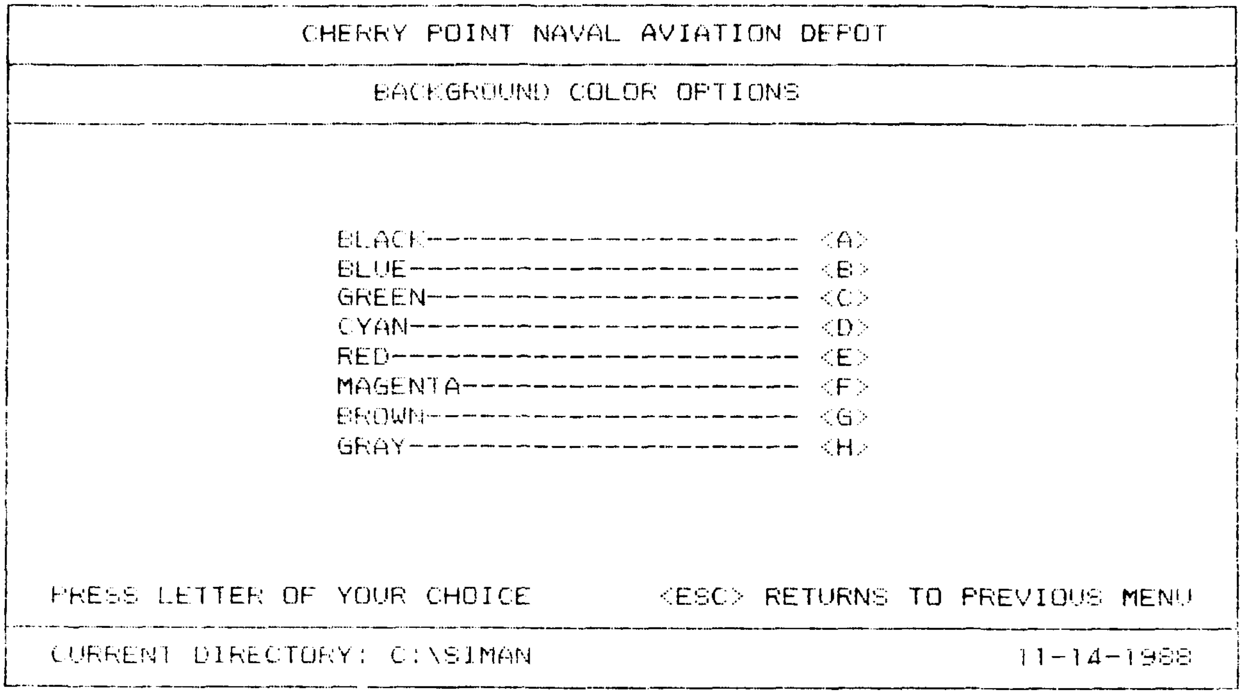
Important Note:

If the user exits to DOS using the DOS gateway he will not be able to load and run Cinema or CSiman since the menu remains in memory, leaving insufficient RAM for these two programs to run. If the user desires to run Cinema or CSiman outside the menu, he must first discontinue use of the menu by pressing Escape at the Main Menu.



```
                 CHERRY POINT NAVAL AVIATION DEPOT

                         MENU OPTIONS MENU



              CHANGE BACKGROUND COLOR--- <A>
              CHANGE FRAME COLOR-------- <B>
              CHANGE TEXT COLOR--------- <C>
              CHANGE BORDER COLOR------- <D>
              CHANGE PRESENT DATE------- <E>



 PRESS LETTER OF YOUR CHOICE        <ESC> RETURNS TO PREVIOUS MENU

 CURRENT DIRECTORY: C:\SIMAN                              11-14-1988
```

This menu provides the user with the ability to change the colors used to display menu options on the screen. It also allows the user to change the date displayed on the screen if for some reason this is desired (i.e., the date was not or incorrectly entered during machine startup.) Selecting one of the color options takes the user to a third-tier screen that provides color options to the user. If option <E> is selected, the user is prompted for the new date. Escape returns the user to the Main Menu.

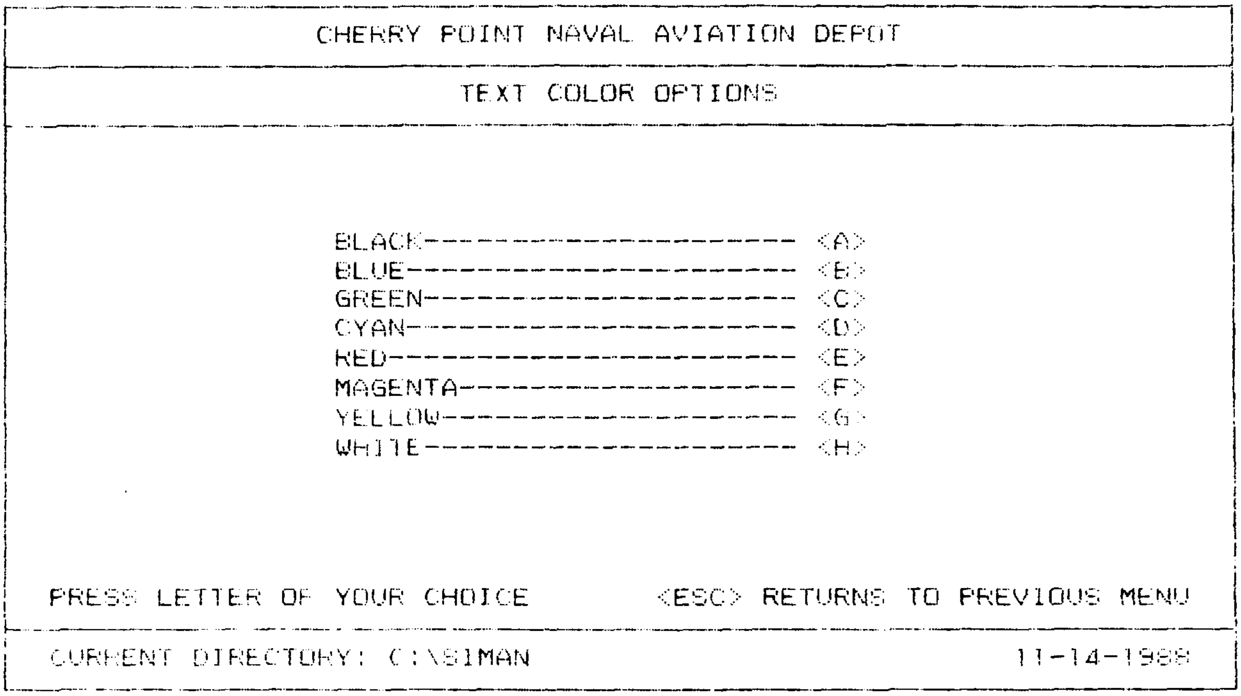


```
                 CHERRY POINT NAVAL AVIATION DEPOT

                     BACKGROUND COLOR OPTIONS



               BLACK--------------------- <A>
               BLUE---------------------- <B>
               GREEN--------------------- <C>
               CYAN---------------------- <D>
               RED----------------------- <E>
               MAGENTA------------------- <F>
               BROWN--------------------- <G>
               GRAY---------------------- <H>



 PRESS LETTER OF YOUR CHOICE      <ESC> RETURNS TO PREVIOUS MENU

 CURRENT DIRECTORY: C:\SIMAN                          11-14-1988
```

This menu screen displays the background color options available. The background color may be changed by pressing the key indicated for the desired color. The option is executed and displayed immediately for user approval. Escape saves the selected background color and returns the user back to the Menu Options Menu.

Important Note:

If the user selects a background color that is the same as the current text color, he is prompted to make another selection since such a combination would cause the text to be invisible.

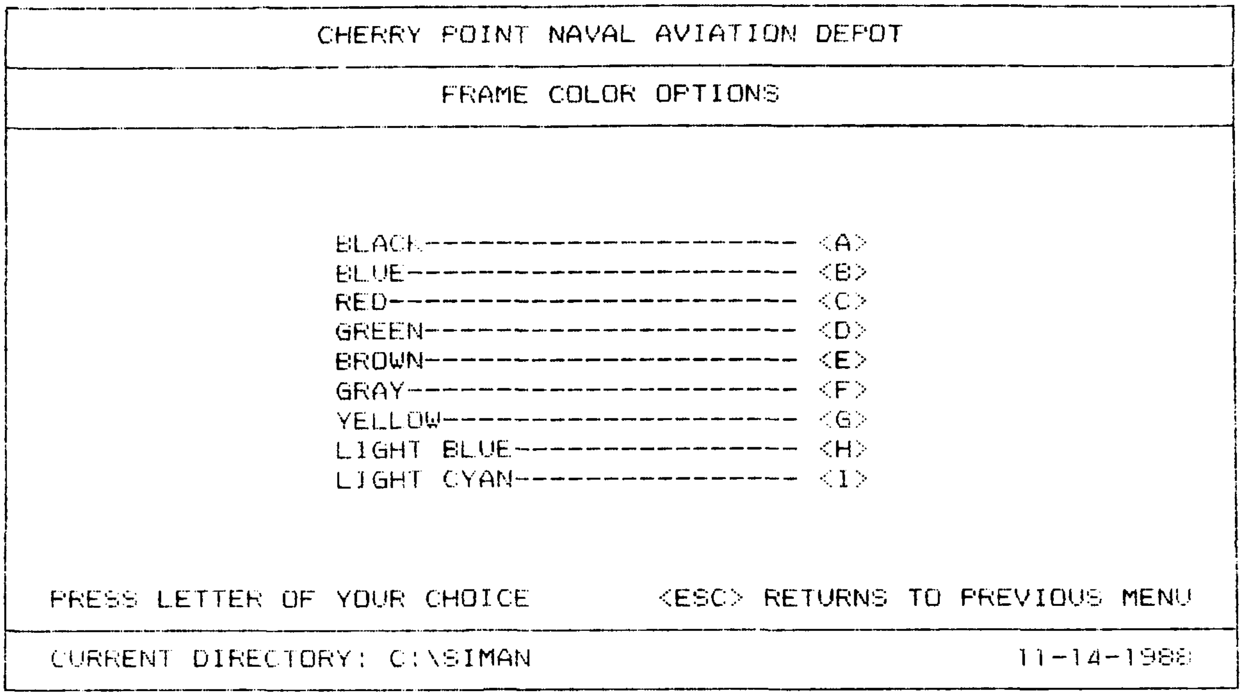


```
                   CHERRY POINT NAVAL AVIATION DEPOT

                          TEXT COLOR OPTIONS


              BLACK---------------------- <A>
              BLUE----------------------- <B>
              GREEN---------------------- <C>
              CYAN----------------------- <D>
              RED------------------------ <E>
              MAGENTA-------------------- <F>
              YELLOW--------------------- <G>
              WHITE---------------------- <H>


  PRESS LETTER OF YOUR CHOICE        <ESC> RETURNS TO PREVIOUS MENU

  CURRENT DIRECTORY: C:\SIMAN                              11-14-1988
```

The above menu provides the user with the options available for the color used to display text on the screen. The color of the text may be changed by pressing the key indicated for the desired color. The option is executed and displayed immediately for user approval. Escape saves the selected text color and returns the user to the Menu Options Menu.

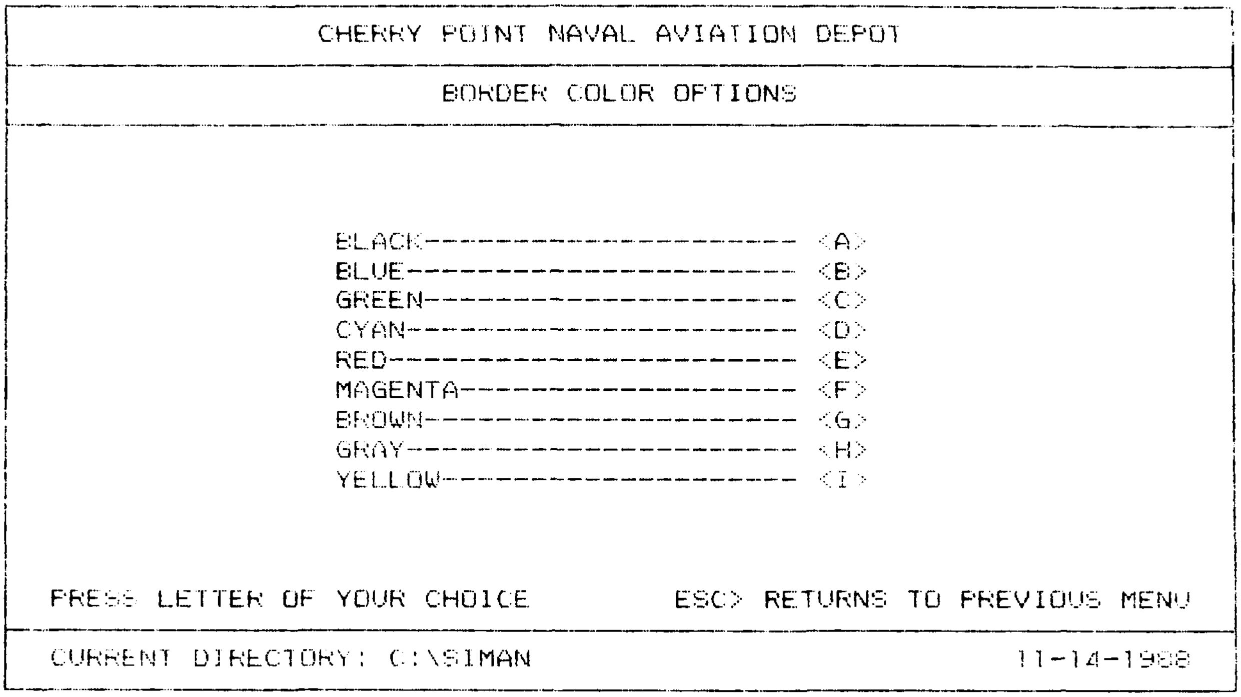
Important Note:

If the user selects a text color that is the same as the current background color, he is prompted to make another selection since such a combination would cause the text to be invisible.



```
                    CHERRY POINT NAVAL AVIATION DEPOT

                          FRAME COLOR OPTIONS


                 BLACK---------------------- <A>
                 BLUE----------------------- <B>
                 RED------------------------ <C>
                 GREEN---------------------- <D>
                 BROWN---------------------- <E>
                 GRAY----------------------- <F>
                 YELLOW--------------------- <G>
                 LIGHT BLUE----------------- <H>
                 LIGHT CYAN----------------- <I>


  PRESS LETTER OF YOUR CHOICE        <ESC> RETURNS TO PREVIOUS MENU

  CURRENT DIRECTORY: C:\SIMAN                              11-14-1988
```

The menu frame refers to the double lines enclosing the menu titles, options, directory, and date. The color of the frame may be changed by pressing the key indicated for the desired color. The option is executed and displayed immediately for user approval. Escape saves the selected frame color and returns the user to the Menu Options Menu



```
                CHERRY POINT NAVAL AVIATION DEPOT
                      BORDER COLOR OPTIONS

                BLACK--------------------- <A>
                BLUE---------------------- <B>
                GREEN--------------------- <C>
                CYAN---------------------- <D>
                RED----------------------- <E>
                MAGENTA------------------- <F>
                BROWN--------------------- <G>
                GRAY---------------------- <H>
                YELLOW-------------------- <I>

 PRESS LETTER OF YOUR CHOICE      ESC> RETURNS TO PREVIOUS MENU
 CURRENT DIRECTORY: C:\SIMAN                          11-14-1988
```

With some video monitors it is possible to change the color of the screen border, the area outside the typical working area of the video display If your monitor allows for this the above options can be used to change the color of the border by pressing the key indicated for the desired color. The option is executed and displayed immediately for user approval. Pressing Escape saves the selected border color and returns the user back to the Menu Options Menu.

Technical Reference

The Menu Files

There are 5 program/data files required to make the menu work. Had CINEMA and CSIMAN not required so much random access memory (RAM), perhaps only one would have been necessary. However, due to memory restrictions imposed by these two programs special attention had to be given to allowing them to run with the menu using as little memory as absolutely necessary.

According to a CINEMA spokesperson, CINEMA and CSIMAN require a minimum of 585K RAM to run. The current DOS RAM limit is 640K. With DOS loaded into RAM, only 595K of this 640K RAM remains for application programs. This allows only 10K for any program loaded into RAM along with CINEMA or CSIMAN (595K - 585K). Again, according to the CINEMA spokesperson, neither program as yet runs under expanded memory.

The compiled menu program, written in BASIC, requires in the neighborhood of 45K RAM (relatively small by most standards but prohibitively large for concurrent use with CINEMA and CSIMAN) A means therefore had to be developed for CINEMA and CSIMAN to be called from, yet run outside, the BASIC main menu (i.e., with the main menu removed from RAM.) Unfortunately, the only way to remove a BASIC program from RAM is to discontinue its use. Catch-22, or so it initially seemed. A little luck and DOS manipulation provided a means around the problem.

The BASIC menuing program will first be explained. A discussion of the DOS batch file used in conjunction with the BASIC program to solve the RAM problem follows that discussion.

The BASIC Menuing Program

The key to understanding how the BASIC menuing program works is in understanding the INKEY and SHELL statements. INKEY provides the means of detecting and identifying keystrokes entered by the user in response to a particular menu option. SHELL provides the pathway to DOS commands, allowing the SIMAN, CINEMA (were it not so large), DOS commmands, or any other executable file to be called and run from within the BASIC program.

All variables are integer and, for convenience in initialization, all variables begin with the letter M. M1 though M10 identify the various menus. M1, the "MAIN MENU", is chosen as the first menu to be displayed upon menu program startup. The screen is cleared and the menu frame is drawn on the screen using the color options read in from the menu data file, NADMENU.DAT, a separate file located in the same directory as the menu program. The appropriate menu options are then retrieved from a subroutine and printed to the screen along with the letter keypress associated with the option. Some calculations are made to ensure centering of the menu title in the menu title box (variable MT, "Menu Title") and centering of the menu options within the menu options box (variable MR, "Menu Row") To complete the menu screen the date and directory subroutine is called. This subroutine displays the date and directory in the date and directory box at the bottom of the screen.

A couple of points should be made: (1) Even though the frame and the text are both foreground colors it is possible to display them on the screen at the same time in different colors by first drawing the frame, changing the foreground color, and then writing the text to the same screen using the new foreground color. (2) The current maximum number of menu options allowed per screen is the default BASIC array size of ten. More than this number can be displayed, however, dimensioning of the menu arrays and different calculations for screen layout would be required.

At this point the program simply waits for input from the user. The inkey routine handles detection and identification of keypresses. As long as no keypress is made, no more commands are processed. INKEY monitors keyboard input and when a keypress is made, the keypress ASCII code, stored by BASIC in INKEY$ is transferred in MC$ ("Menu Choice"). Option Q, the tenth option in every case, is reserved for "Quit", the action which returns the user to the previous menu or exits the program if the "Main Menu" screen is displayed. For convenience, the Escape key has been defined to perform the same functions since most users are accustomed to using this key to perform such operations.

The Keystroke Definitions routine defines what line number to proceed to upon each given keypress. The statements allow for both upper and lower case. If an illegal keypress is detected, the statements cause an audible signal to be produced and program execution proceeds back to the inkey routine for another keypress. The space before the first letter ("_AaBb...) is required since it represents in this syntax everything other than a valid keypress. The line numbers at the end of these statements correspond to its respective keypress listed within the parentheses

At this point it is simply a matter of writing the commands at the listed line number necessary to carry out the desired actions If the desired action is something that can be carried out by BASIC, the BASIC commands are presented. If the desired action requires a DOS command, the SHELL statement is used with the desired DOS commands enclosed in quotes.

The DOS Batch File Addition to the BASIC Menuing Program

As mentioned earlier, the BASIC program described above, upon compilation, is too large to run concurrently with CINEMA and CSIMAN. Batch file options were therefore considered for this purpose.

There were two problems encounterd when trying to determine how to use batch files for menuing. The first problem, really an anoyance, is that since batch files read and execute one statement at a time they tend to be very slow. Each command statement must be read from the disk prior to execution--an operation significantly slower than the time required to execute statements already resident in RAM. The second, more imposing problem is that DOS did not come with a utility for keystroke detection, per se. The only way of providing input to a batch file is through user-provided parameters included when the batch file is initially executed or through pause statements which simply prompt the user to strike a key when ready to continue. Neither of these lend themselves readily to the use of multiple option situations. Even if it were possible to imitate the BASIC menu screen using a DOS batch file (and it is), there was no interactive way available to determine if a key had been pressed in response to an option, much less which key it was.

Fortunately, with the aid of an article I read, I was able to create a DOS utility (command file) capable of recognizing keystrokes in batch files. (PC WORLD, October 1988) The program, "GETKEY.COM" is a small command file that detects and returns the ASCII code of the key being pressed. This code can then be used as the parameter in the DOS IF statement's ERRRORLEVEL condition (refer to DOS manual IF statement for batch files). It then becomes simply a logic problem to determine which key has been pressed.

The IF ERRORLEVEL = NUMBER statement executes only if the previously executed command has an exit code of NUMBER or higher. In the case of GETKEY.COM, the exit code NUMBER is the ASCII code of the key being pressed. Since ERRORLEVEL operates under the above inequality condition, care must be taken to ensure that a single ASCII code causes the desired reaction (i.e., X, Y, and Z, ASCII codes 88, 89, and 90 respectively all satisfy the ERRORLEVEL condition ERRORLEVEL = 88). Simply put, the IF ERRORLEVEL = NUMBER statement should treated as an IF ERRORLEVEL >= NUMBER statement.

Two tests are therefore required within the batch file to detect a specific keypress. A check must first be made for the ASCII code of the next higher character, eliminating it and all others above it from consideration before checking for the ASCII code of a valid keypress. It follows then that when checking for multiple options, tests must be arranged in decreasing ASCII order. All this ensures, for example, that a <B> is not mistaken for an <A>.

Having provided a means for keypress detection in batch files it became possible to create a batch file that behaved similar to the BASIC menu. In fact, it is actually possible to have the BASIC program do everything to the point of waiting for the keypress, including displaying the screen of options. (The batch file, as mentioned above, can recreate the options screen, but does so at a noticeably slower pace--one line at a time.)

Since it is necessary to exit the BASIC program completely and use the much smaller batch file (approximately 600 bytes) to use the animation programs, the shell command within the BASIC program could not be used to call the batch file. Therefore, the batch file has to begin execution of the BASIC menu program so that when the BASIC program is terminated the batch file automatically resumes control. When the CINEMA Menu option is selected from within the BASIC program, the BASIC program simply draws the menu options screen and terminates. The batch file then takes over and handles keypress detection for CINEMA/CSIMAN startup.

Since it makes sense that menu termination be a Main Menu option, a trick had to be devised to identify to the batch file whether control is being returned from the terminated BASIC program to allow CINEMA options to be processed or if the BASIC program is actually being terminated because the user is through with the menu. To handle this, the BASIC program creates a "dummy" data file if the exit option returning control to the batch file is a true menu termination. When control returns to the batch file, the IF EXIST "filename" test checks to see if "stoptest.dat" exists. If so, the batch file deletes this file and terminates. The user is returned to the root directory DOS prompt and the menu is completely stopped and must be restarted by typing "GOMENU". On the other hand, if the BASIC program is being terminated and control is sent back to the batch file to allow for CINEMA and CSIMAN execution, the BASIC menu program does not create "stoptest.bat" and simply terminates. When the batch file does not find this file it prints the CINEMA options screen and waits for the user to enter an option selection. Upon completion of a CINEMA or CSIMAN session, the GOCINEMA and GOCSIMAN batch files recall the BASIC program and the Main Menu is restarted. Pressing the escape key also restarts the Main Menu. Once back to the Main Menu the user may continue using the menu or press escape to exit.

Files Required to Run the Menu

Filename	Purpose
Nadmenu.exe	Contains all menu options/commands except CINEMA.
Gomenu.bat	Calls Nadmenu.exe. Controls CINEMA options.
Getkey.com	Provides keypress detection for Gomenu.bat.
Beep.com	Creates audible signal. Necessary for Gomenu.bat.
Nadmenu.dat	Storage location for color/directory data.

Directory Names Assumed by the Menu

Directory	Contents
C:\	Root Directory
C:\CINEMA	CINEMA Executable Program Files
C:\CINEMA\ANIMATE	CINEMA Data Files
C:\SIMAN	Siman Executable Program Files
C:\SIMAN\LEVEL1	Level 1 Simulation Data Files
C:\SIMAN\LEVEL2	Level 2 Simulation Data Files
C:\SIMAN\LEVEL3	Level 3 Simulation Data Files
C:\NADLP	Menu, MenuLP, and Associated Data Files
C:\MOUSE	Mouse Driver. (Needed for CINEMA)
C:\NADMENU	Above Listed Menu Files

The NADMENU.BAS Basic Program Listing

```
1 ' **************************** INITIALIZE ****************************
2 '
10 CLEAR:DEFINT M
20 M1=1:M2=0:M3=0:M4=0:M5=0:M6=0:M7=0:M8=0:M9=0:M10=0:M11=0
30 CHDIR "\NADMENU":OPEN "NADMENU.DAT" FOR INPUT AS #1
40 INPUT
#1,MTEXT#,MBACK#,MBORDER#,MFRAME#,PATH$,LAYFILE$,PROGFILE$:CLOSE
1:CHDIR PATH$
45 '
50 ' ******** RETRIEVE APPROPRIATE MENU OPTIONS DESCRIPTIONS *********
55 '
60 IF M1 THEN GOSUB 11000
65 IF M2 THEN GOSUB 12000
70 IF M3 THEN GOSUB 13000
75 IF M4 THEN GOSUB 14000
80 IF M5 THEN GOSUB 15000
85 IF M6 THEN GOSUB 16000
90 IF M7 THEN GOSUB 17000
95 IF M8 THEN GOSUB 18000
100 IF M9 THEN GOSUB 19000
105 IF M10 THEN GOSUB 20000
110 IF M11 THEN GOSUB 21000
115 '
120 ' ****** READ IN TEXT, BACKGROUND, BORDER, AND FRAME COLORS ******
125 '
130 COLOR MFRAME#,MBACK#,MBORDER#:CLS:GOSUB 40000
140 COLOR MTEXT#,MBACK#.MBORDER#
150 '190 '
200 ' ************************ PRINT  MENU TEXT ************************
205 '
210 MR=(25-7-MN)/2:MT=LEN(MT$):MT=(80-MT)/2
220 LOCATE 3,24:PRINT "CHERRY POINT NAVAL AVIATION DEPOT"
230 LOCATE 5,MT:PRINT MT$:MR=MR+2
240 FOR MY=1 TO MN:MR=MR+1
250 LTR$=MID$("ABCDEFGHIQ",MY,1)
260 ML=LEN(MA$(MY))
270 LOCATE MR+1,24:PRINT MA$(MY):"";STRING$(27-ML,"-");" <"LTR$">";
280 NEXT MY
290 LOCATE 20,5
300 IF M1 THEN PRINT "PRESS LETTER OF YOUR CHOICE
   <ESC> EXITS TO DOS":GOTO 320
305 IF M10 THEN PRINT "PRESS LETTER OF YOUR CHOICE
<ESC> RETURNS TO MAIN MENU":GOTO 320
310 PRINT "PRESS LETTER OF YOUR CHOICE              <ESC> RETURNS TO
PREVIOUS MENU"
320 GOSUB 30000
330 IF M10 GOTO 9100
340 GOTO 500
450 BEEP
470 '
500 ' **************************** INKEY ROUTINE ****************************
505 '
510 MC$="":WHILE MC$="":MC$=INKEY$:WEND
520 IF MC$=CHR$(027) THEN MC$="Q"
530 '
```

```
600 ' ********************* KEYSTROKE DEFINITIONS ********************
605 '
610 IF M1 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1110,1110,1120,1120,1130,1130,1140,1140,1150,1150,1160,1160,1170,1
170,1190,1180,1190,1190,1195,1195
620 IF M2 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1210,1210,1220,1220,1230,1230,1240,1240,1250,1250,1260,1260,1270,1
270,1280,1280,1290,1290,1295,1295
630 IF M3 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1310,1310,1320,1320,1330,1330,1340,1340,1350,1350,1360,1360,1370,1
370,1380,1380,1390,1390,1395,1395
640 IF M4 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1410,1410,1420,1420,1430,1430,1440,1440,1450,1450,1460,1460,1470,1
470,1480,1480,1490,1490,1495,1495
650 IF M5 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1510,1510,1520,1520,1530,1530,1540,1540,1550,1550,1560,1560,1570,1
570,1580,1580,1590,1590,1595,1595
660 IF M6 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1610,1610,1620,1620,1630,1630,1640,1640,1650,1650,1660,1660,1670,1
670,1680,1680,1690,1690,1695,1695
670 IF M7 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1710,1710,1720,1720,1730,1730,1740,1740,1750,1750,1760,1760,1770,1
770,1780,1760,1790,1790,1795,1795
680 IF M8 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1810,1810,1820,1820,1830,1830,1840,1840,1850,1850,1860,1960,1870,1
870,1880,1880,1890,1890,1895,1895
690 IF M9 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,1910,1910,1920,1920,1930,1930,1940,1940,1950,1950,1960,1960,1970,1
970,1980,1980,1990,1990,1995,1995
700 IF M11 THEN ON 1+INSTR(" AaBbCcDdEeFfGgHhIiQq",MC$) GOTO  450,
450,2110,2110,2120,2120,2130,2130,2140,2140,2150,2150,2160,2160,2170,2
170,2180,2180,2190,2190,2195,2195
1095 '
1100 ' ********************* MAIN MENU COMMANDS ********************
1105 '
1110 GOTO 8200
1120 GOTO 8300
1130 GOTO 8960
1140 GOTO 8400
1150 GOTO 8500
1160 GOTO 450
1170 GOTO 450
1180 GOTO 450
1190 GOTO 450
1195 GOTO 9000
1199 '
1200 ' ********************* SIMULATION MENU COMMANDS *****************
1205 '
1210 TIMER OFF:CLS:CHDIR "C:\SIMAN\LEVEL1":SHELL "GOSIMAN
LEVEL1":GOSUB 10020:GOTO 50
1220 TIMER OFF:CLS:CHDIR "C:\SIMAN\LEVEL2":SHELL "GOSIMAN
LEVEL2":GOSUB 10020:GOTO 50
1230 GOTO 450
1240 TIMER OFF:CLS:CHDIR "C:\SIMAN\LEVEL1":SHELL
"MORE<LEVEL1.OUT":GOSUB 10020:GOTO 50
1250 TIMER OFF:CLS:CHDIR "C:\SIMAN\LEVEL2":SHELL
"MORE<LEVEL2.OUT":GOSUB 10020:GOTO 50
1260 GOTO 450
1270 LOCATE 18,25:COLOR M1EXT#+16:PRINT "PREPARING LEVEL 2 FOR
ANIMATION":SHELL "COPY C:\SIMAN\LEVEL2\LEVEL2.P
```

```
C:\CINEMA\ANIMATE":GOTO 50
1280 GOTO 450
1290 GOTO 450
1295 GOTO 8100
1299 '
1300 ' ********************* LP MENU COMMANDS **************************
1305 '
1310 TIMER OFF:CLS:SHELL "CD C:\NADLP":SHELL "MENU":GOSUB 10020:GOTO
50
1320 TIMER OFF:CLS:SHELL "CD C:\NADLP":SHELL "MENULP":GOSUB 10020:GOTO
50
1330 TIMER OFF:CLS:CHDIR "C:\NADLP":SHELL "MORE<BLADES.DAT":GOSUB
10020:CHDIR PATH$:GOTO 50
1340 TIMER OFF:CLS:CHDIR "C:\NADLP":SHELL "MORE<CALENDAR.DAT":GOSUB
10020:CHDIR PATH$:GOTO 50
1350 TIMER OFF:CLS:CHDIR "C:\NADLP":SHELL "MORE<INPUT.DAT":GOSUB
10020:CHDIR PATH$:GOTO 50
1360 TIMER OFF:CLS:CHDIR "C:\NADLP":SHELL "MORE<MACHINES.DAT":GOSUB
10020:CHDIR PATH$:GOTO 50
1370 GOTO 450
1380 GOTO 450
1390 GOTO 450
1395 GOTO 8100
1399 '
1400 ' ********************** DOS MENU COMMANDS **********************
1405 '
1410 CLS:LOCATE 5,23:PRINT "IN DOS, TYPE EXIT TO RETURN TO
MENU":LOCATE 10,25:INPUT "PRESS ENTER TO PROCEED TO DOS ",D
1415 TIMER OFF:CLS:SHELL:GOTO 50
1420 TIMER OFF:CLS:FILES:GOSUB 10020:GOTO 50
1430 TIMER OFF:CLS:LOCATE 10,20:PRINT "THE CURRENT DIRECTORY IS:
"PATH$:LOCATE 15,20:INPUT "ENTER NEW DIRECTORY: "PATH$:ON ERROR GOTO
1433:GOTO 1435
1433 BEEP:LOCATE 20,15:INPUT "DIRECTORY NOT FOUND, PLEASE RE-ENTER:
"PATH$:RESUME:GOTO 1430
1435 CHDIR PATH$:GOTO 50
1440 GOTO 450
1450 GOTO 450
1460 GOTO 450
1470 GOTO 450
1480 GOTO 450
1490 GOTO 450
1495 GOTO 8100
1499 '
1500 ' ******************** MENU OPTIONS COMMANDS ********************
1505 '
1510 GOTO 8600
1520 GOTO 8700
1530 GOTO 8800
1540 GOTO 8900
1550 TIMER OFF:CLS:SHELL "DATE":GOTO 50
1560 GOTO 450
1570 GOTO 450
1580 GOTO 450
1590 GOTO 450
1595 GOTO 8100
1599 '
1600 ' ************** BACKGROUND COLOR OPTIONS COMMANDS **************
1605 '
1610 MBACKTMP=MBACK#:MBACK#=0:GOTO 50
```

```
1620 MBACKTMP=MBACK#:MBACK#=1:GOTO 50
1630 MBACKTMP=MBACK#:MBACK#=2:GOTO 50
1640 MBACKTMP=MBACK#:MBACK#=3:GOTO 50
1650 MBACKTMP=MBACK#:MBACK#=4:GOTO 50
1660 MBACKTMP=MBACK#:MBACK#=5:GOTO 50
1670 MBACKTMP=MBACK#:MBACK#=6:GOTO 50
1680 MBACKTMP=MBACK#:MBACK#=7:GOTO 50
1690 GOTO 450
1695 GOTO 8500
1699 '
1700 ' **************** FRAME COLOR OPTIONS COMMANDS *****************
1705 '
1710 MFRAME#=0:GOSUB 40000:GOTO 500
1720 MFRAME#=1:GOSUB 40000:GOTO 500
1730 MFRAME#=2:GOSUB 40000:GOTO 500
1740 MFRAME#=3:GOSUB 40000:GOTO 500
1750 MFRAME#=4:GOSUB 40000:GOTO 500
1760 MFRAME#=5:GOSUB 40000:GOTO 500
1770 MFRAME#=14:GOSUB 40000:GOTO 500
1780 MFRAME#=15:GOSUB 40000:GOTO 500
1790 GOTO 450
1795 GOTO 8500
1799 '
1800 ' ***************** TEXT COLOR OPTIONS COMMANDS *****************
1805 '
1810 MTEXTTMP=MTEXT#:MTEXT#=0:GOTO 50
1820 MTEXTTMP=MTEXT#:MTEXT#=1:GOTO 50
1830 MTEXTTMP=MTEXT#:MTEXT#=2:GOTO 50
1840 MTEXTTMP=MTEXT#:MTEXT#=3:GOTO 50
1850 MTEXTTMP=MTEXT#:MTEXT#=4:GOTO 50
1860 MTEXTTMP=MTEXT#:MTEXT#=5:GOTO 50
1870 MTEXTTMP=MTEXT#:MTEXT#=14:GOTO 50
1880 MTEXTTMP=MTEXT#:MTEXT#=15:GOTO 50
1890 GOTO 450
1895 GOTO 8500
1899 '
1900 ' **************** BORDER COLOR OPTIONS COMMANDS ****************
1905 '
1910 MBORDER#=0:GOTO 50
1920 MBORDER#=1:GOTO 50
1930 MBORDER#=2:GOTO 50
1940 MBORDER#=3:GOTO 50
1950 MBORDER#=4:GOTO 50
1960 MBORDER#=5:GOTO 50
1970 MBORDER#=6:GOTO 50
1980 MBORDER#=7:GOTO 50
1990 MBORDER#=14:GOTO 50
1995 GOTO 8500
1999 '
2100 ' **************** SELECT ANIMATION FILE COMMANDS ***************
2105 '
2110 GOTO 8950
2120 CLS:GOSUB 55000:GOTO 50
2130 GOTO 450
2140 GOTO 450
2150 GOTO 450
2160 GOTO 450
2170 GOTO 450
2180 GOTO 450
2190 GOTO 450
```

```
2195 GOTO 8100
8000 '
8010 ' ******************* MENU SELECTION ROUTINES *******************
8020 '
8100 CLS:M1=1:M2=0:M3=0:M4=0:M5=0:M6=0:M7=0:M8=0:M9=0:M10=0:M11=0:GOTO
50
8200 CLS:M1=0:M2=1:M3=0:M4=0:M5=0:M6=0:M7=0:M8=0:M9=0:M10=0:M11=0:GOTO
50
8300 CLS:M1=0:M2=0:M3=1:M4=0:M5=0:M6=0:M7=0:M8=0:M9=0:M10=0:M11=0:GOTO
50
8400 CLS:M1=0:M2=0:M3=0:M4=1:M5=0:M6=0:M7=0:M8=0:M9=0:M10=0:M11=0:GOTO
50
8500 CLS:M1=0:M2=0:M3=0:M4=0:M5=1:M6=0:M7=0:M8=0:M9=0:M10=0:M11=0:GOTO
50
8600 CLS:M1=0:M2=0:M3=0:M4=0:M5=0:M6=1:M7=0:M8=0:M9=0:M10=0:M11=0:GOTO
50
8700 CLS:M1=0:M2=0:M3=0:M4=0:M5=0:M6=0:M7=1:M8=0:M9=0:M10=0:M11=0:GOTO
50
8800 CLS:M1=0:M2=0:M3=0:M4=0:M5=0:M6=0:M7=0:M8=1:M9=0:M10=0:M11=0:GOTO
50
8900 CLS:M1=0:M2=0:M3=0:M4=0:M5=0:M6=0:M7=0:M8=0:M9=1:M10=0:M11=0:GOTO
50
8950 CLS:M1=0:M2=0:M3=0:M4=0:M5=0:M6=0:M7=0:M8=0:M9=0:M10=1:M11=0:GOTO
50
8960 CLS:M1=0:M2=0:M3=0:M4=0:M5=0:M6=0:M7=0:M8=0:M9=0:M10=0:M11=1:GOTO
50
9000 '
9005 ' ****************** END BASIC PROGRAM ROUTINE ******************
9010 '
9020 CHDIR "\NADMENU".SHELL "DEL NADMENU.DAT":OPEN "NADMENU.DAT" FOR
OUTPUT AS #2
9030 WRITE #2, MTEXT#;MBACK#;MBORDER#;MFRAME#;PATH$;LAYFILE$;PROGFILE$
9040 OPEN "STOPTEST.DAT" FOR OUTPUT AS #3
9050 CLOSE 1:CLOSE 2:CLOSE 3:END
9060 '
9100 ' ******** RETURN TO BATCH FILE/RUN CINEMA MENU ROUTINE *********
9105 '
9210 CHDIR "\NADMENU":SHELL "DEL NADMENU.DAT":OPEN "NADMENU.DAT" FOR
OUTPUT AS #2
9220 WRITE #2, MTEXT#;MBACK#;MBORDER#;MFRAME#;PATH$;LAYFILE$;PROGFILE$
9230 CHDIR "\NADMENU":END
9240 '
10000 ' *********************** INKEY ROUTINE ************************
10005 '
10010 LOCATE 25, 5:PRINT "PRESS ANY KEY TO RETURN TO MENU":GOTO 10030
10020 LOCATE 25,24:PRINT "PRESS ANY KEY TO RETURN TO MENU"
10030 MD$="":WHILE MD$="":MD$=INKEY$:WEND:CLS:RETURN
10040 '
11000 ' *********************** MAIN MENU DATA ***********************
11005 '
11010 MT$="MAIN MENU":MN=5:REM   MT=MENU TITLE   MN=# MENU OPTIONS
11020 MA$(1)=" GO TO SIMULATION MENU":
11030 MA$(2)=" GO TO LP MENU":
11040 MA$(3)=" GO TO CINEMA MENU":
11050 MA$(4)=" GO TO DOS MENU":
11060 MA$(5)=" GO TO MENU OPTIONS MENU":
11070 MA$(6)=" ":
11080 MA$(7)=" ":
11090 MA$(8)=" ":
11100 MA$(9)=" ":
```

```
11110 MA$(10)=" RETURN TO ENTRY LEVEL MENU":
11120 RETURN
11130 '
12000 ' ******************** SIMULATION MENU DATA ********************
12005 '
12010 MT$="SIMULATION MENU":MN=8
12020 MA$(1)=" SIMULATE EDITED LEVEL 1":
12030 MA$(2)=" SIMULATE EDITED LEVEL 2":
12040 MA$(3)=" SIMULATE EDITED LEVEL 3":
12050 MA$(4)=" SHOW LEVEL 1 OUTPUT":
12060 MA$(5)=" SHOW LEVEL 2 OUTPUT":
12070 MA$(6)=" SHOW LEVEL 3 OUTPUT":
12080 MA$(7)=" PREP LEVEL 2 FOR ANIMATION":
12090 MA$(8)=" PREP LEVEL 3 FOR ANIMATION":
12100 MA$(9)=" ":
12110 MA$(10)=" RETURN TO MAIN MENU":
12120 RETURN
12130 '
13000 ' **************** LINEAR PROGRAMMING MENU DATA ****************
13005 '
13010 MT$="LINEAR PROGRAMMING MENU":MN=6
13020 MA$(1)=" RUN MENU":
13030 MA$(2)=" RUN MENU LP":
13040 MA$(3)=" SHOW BLADES.DAT":
13050 MA$(4)=" SHOW CALENDAR.DAT":
13060 MA$(5)=" SHOW INPUT.DAT":
13070 MA$(6)=" SHOW MACHINES.DAT":
13080 MA$(7)=" ":
13090 MA$(8)=" ":
13100 MA$(9)=" ":
13110 MA$(10)=" RETURN TO MAIN MENU":
13120 RETURN
13130 '
14000 ' *********************** DOS MENU DATA ************************
14005
14010 MT$="DOS MENU":MN=3
14020 MA$(1)=" EXIT TO DOS":
14030 MA$(2)=" SHOW DIRECTORY FILES":
14040 MA$(3)=" CHANGE CURRENT DIRECTORY":
14050 MA$(4)=" ":
14060 MA$(5)=" ":
14070 MA$(6)=" ":
14080 MA$(7)=" ":
14090 MA$(8)=" ":
14100 MA$(9)=" ":
14110 MA$(10)=" RETURN TO MAIN MENU":
14120 RETURN
14130 '
15000 ' ******************* MENU OPTIONS MENU DATA *******************
15005 '
15010 MT$="MENU OPTIONS MENU":MN=5
15020 MA$(1)=" CHANGE BACKGROUND COLOR":
15030 MA$(2)=" CHANGE FRAME COLOR":
15040 MA$(3)=" CHANGE TEXT COLOR":
15050 MA$(4)=" CHANGE BORDER COLOR":
15060 MA$(5)=" CHANGE PRESENT DATE":
15070 MA$(6)=" ":
15080 MA$(7)=" ":
15090 MA$(8)=" ":
15100 MA$(9)=" ":
```

```
15110 MA$(10)=" RETURN TO MAIN MENU":
15120 RETURN
15130 '
16000 ' ***************** BACKGROUND COLOR MENU DATA *****************
16005 '
16010 MT$="BACKGROUND COLOR OPTIONS":MN=8
16020 MA$(1)=" BLACK":
16030 MA$(2)=" BLUE":
16040 MA$(3)=" GREEN":
16050 MA$(4)=" CYAN":
16060 MA$(5)=" RED":
16070 MA$(6)=" MAGENTA":
16080 MA$(7)=" BROWN":
16090 MA$(8)=" GRAY":
16100 MA$(9)=" ":
16110 MA$(10)=" RETURN TO MENU OPTIONS":
16120 RETURN
16130 '
17000 ' ******************* FRAME COLORS MENU DATA *******************
17005 '
17010 MT$="FRAME COLOR OPTIONS":MN=8
17020 MA$(1)=" BLACK":
17030 MA$(2)=" BLUE":
17040 MA$(3)=" GREEN":
17050 MA$(4)=" CYAN":
17060 MA$(5)=" RED":
17070 MA$(6)=" MAGENTA":
17080 MA$(7)=" YELLOW":
17090 MA$(8)=" WHITE":
17100 MA$(9)=" ":
17110 MA$(10)=" RETURN TO MENU OPTIONS":
17120 RETURN
17130 '
18000 ' ******************* TEXT COLORS MENU DATA ********************
18005 '
18010 MT$="TEXT COLOR OPTIONS":MN=8
18020 MA$(1)=" BLACK":
18030 MA$(2)=" BLUE":
18040 MA$(3)=" GREEN":
18050 MA$(4)=" CYAN":
18060 MA$(5)=" RED":
18070 MA$(6)=" MAGENTA":
18080 MA$(7)=" YELLOW":
18090 MA$(8)=" WHITE":
18100 MA$(9)=" ":
18110 MA$(10)=" RETURN TO MENU OPTIONS":
18120 RETURN
18130 '
19000 ' ******************* BORDER COLORS MENU DATA ******************
19005 '
19010 MT$="BORDER COLOR OPTIONS":MN=9
19020 MA$(1)=" BLACK":
19030 MA$(2)=" BLUE":
19040 MA$(3)=" GREEN":
19050 MA$(4)=" CYAN":
19060 MA$(5)=" RED":
19070 MA$(6)=" MAGENTA":
19080 MA$(7)=" BROWN":
19090 MA$(8)=" GRAY":
19100 MA$(9)=" YELLOW":
```

```
19110 MA$(10)=" RETURN TO MENU OPTIONS":
19120 RETURN
19130 '
20000 ' ********************** CINEMA MENU DATA **********************
20005 '
20010 MT$="CINEMA MENU":MN=6
20020 MA$(1)=" EDIT LEVEL 2 LAYOUT":
20030 MA$(2)=" ANIMATE LEVEL 2":
20040 MA$(3)=" EDIT LEVEL 3 LAYOUT":
20050 MA$(4)=" ANIMATE LEVEL 3":
20060 MA$(5)=" EDIT "+LAYFILE$+" LAYOUT":
20070 MA$(6)=" ANIMATE "+PROGFILE$:
20080 MA$(7)=" ":
20090 MA$(8)=" ":
20100 MA$(9)=" ":
20110 MA$(10)=" ":
20120 RETURN
20130 '
21000 ' ******************** SELECT ANIMATION FILE DATA **************
21005 '
21010 MT$="CINEMA MENU":MN=2
21020 MA$(1)=" GO TO ANIMATION MENU":
21030 MA$(2)=" CHANGE USER-DEFINED FILE":
21040 MA$(3)=" ":
21050 MA$(4)=" ":
21060 MA$(5)=" ":
21070 MA$(6)=" ":
21080 MA$(7)=" ":
21090 MA$(8)=" ":
21100 MA$(9)=" ":
21110 MA$(10)=" ":
21120 RETURN
21130 '
30000 ' ***************** DATE/DIRECTORY SUBROUTINE ******************
30005 '
30110 LOCATE 22,5:PRINT "CURRENT DIRECTORY IS: "PATH$
30120 LOCATE 22,65:PRINT DATE$:RETURN
30130 '
40000 ' *********************** SCREEN BORDER ************************
40005 '
40008 IF MTEXT#=MBACK# THEN GOSUB 50000
40010 COLOR MFRAME#,MBACK#,MBORDER#
40020 LOCATE 2,2:PRINT STRING$(1,CHR$(201)); STRING$(75,CHR$(205));
STRING$(1,CHR$(187))
40030 FOR MROW=3 TO 23
40040 LOCATE MROW,2:PRINT STRING$(1,CHR$(186)):LOCATE MROW,78:PRINT
STRING$(1,186)
40050 NEXT MROW
40060 LOCATE 23,2:PRINT STRING$(1,CHR$(200)); STRING$(75,CHR$(205));
STRING$(1,CHR$(188))
40070 LOCATE 4,2:PRINT STRING$(1,CHR$(204)); STRING$(75,CHR$(205));
STRING$(1,CHR$(185))
40080 LOCATE 6,2:PRINT STRING$(1,CHR$(204)); STRING$(75,CHR$(205));
STRING$(1,CHR$(185))
40090 LOCATE 21,2:PRINT STRING$(1,CHR$(204)); STRING$(75,CHR$(205));
STRING$(1,CHR$(185))
40100 IF M11 THEN LOCATE 17,15:PRINT "CURRENT USER-DEFINED ANIMATION
LAYOUT FILE IS: "LAYFILE$
40110 IF M11 THEN LOCATE 18,15:PRINT "CURRENT USER-DEFINED ANIMATION
PROGRAM FILE IS: "PROGFILE$
```

```
40150 RETURN
50000 IF M8 THEN MTEXT#=MTEXTTMP
50010 IF M6 THEN MBACK#=MBACKTMP
50020 COLOR MTEXT#,MBACK#,MBORDER#:CLS
50030 LOCATE 18,11:PRINT "SELECTED TEXT WOULD NOT BE VISIBLE ON
SELECTED BACKGROUND"
50040 LOCATE 19,25:PRINT "PLEASE MAKE ANOTHER SELECTION"
50050 RETURN
50060 '
55000 ' ************ USER-DEFINED ANIMATION FILE SUBROUTINE **********
55005 '
55010 CLS:GOSUB 40000:GOSUB 30000
55020 LOCATE 3,24:PRINT "CHERRY POINT NAVAL AVIATION DEPOT"
55030 LOCATE 5,30:PRINT "DEFINE USER ANIMATION"
55040 PROGFILE$="":LAYFILE$=""
55050 LOCATE 8,5
55060 INPUT "NAME OF USER-DEFINED ANIMATION LAYOUT FILE (NO
EXTENSION)";LAYFILE$
55070 LOCATE 9,5
55080 INPUT "NAME OF USER-DEFINED ANIMATION PROGRAM FILE (NO
EXTENSION)";PROGFILE$
55150 LOCATE 11,15:PRINT "LAYOUT FILE NAME IS --------------- "
LAYFILE$+".LAY"
55160 LOCATE 13,15:PRINT "PROGRAM FILE NAME IS --------------- "
PROGFILE$+".P"
55170 LOCATE 15,33:INPUT "IS THIS CORRECT";ANS$
55180 IF ANS$="YES" THEN GOTO 55210
55190 IF ANS$="Y" THEN GOTO 55210
55200 GOTO 55000
55210 CHDIR "\CINEMA\ANIMATE":OPEN "GOEDLAY.BAT" FOR OUTPUT AS #4:OPEN
"GOANIM.BAT" FOR OUTPUT AS #5
55220 PRINT #4, "GOCINEMA",LAYFILE$
55230 PRINT #5, "GOCSIMAN",LAYFILE$,PROGFILE$
55240 CLOSE 4:CLOSE 5
55250 RETURN
```

NADMENU.DAT Data Descriptions

```
15,5,5,15,"C:\SIMAN","ORANGE"
^  ^ ^  ^     ^         ^
¦  ¦ ¦  ¦     ¦         ¦________ Current User-Defined Animation File
¦  ¦ ¦  ¦     ¦___________________ Current Directory
¦  ¦ ¦  ¦__________________________ Current Frame Color
¦  ¦ ¦______________________________ Current Border Color
¦  ¦_________________________________ Current Background Color
¦_____________________________________ Current Text Color
```

The GOMENU.BAT Batch File

```
echo off
if exist stoptest.dat then del stoptest.dat
nadmenu
if exist stoptest.dat goto exit
:getkey
getkey
if errorlevel 103 goto error
if errorlevel 102 goto csimanud
if errorlevel 101 goto cinemaud
if errorlevel 100 goto csiman13
if errorlevel 99 goto cinema13
if errorlevel 98 goto csiman12
if errorlevel 97 goto cinema12
if errorlevel 71 goto error
if errorlevel 70 goto csimanud
if errorlevel 69 goto cinemaud
if errorlevel 68 goto csiman13
if errorlevel 67 goto cinema13
if errorlevel 66 goto csiman12
if errorlevel 65 goto cinema12
if errorlevel 28 goto error
if errorlevel 27 goto menu
:error
beep
goto getkey
:menu
chdir\nadmenu
nadmenu
if exist stoptest.dat goto exit
goto getkey
:cinema12
chdir\cinema
gocinema level2
goto getkey
:csiman12
chdir\cinema
gocsiman level2
goto getkey
:cinema13
chdir\cinema
gocinema level3
goto getkey
:csiman13
chdir\cinema
gocsiman level3
goto getkey
:cinemaud
chdir\cinema
goedlay
goto getkey
:csimanud
chdir\cinema
goanim
goto getkey
:exit
del stoptest.dat
chdir\
cls
```

Attachment 1

Sample Level 1 Simulation Files

SIMAN Variable Dictionary: Level1

Attribute #	Description
1	Station service time -- updated automatically for each station in sequence
2-30	Wait time in machine queues at stations 1-29 respectively
31	Batch arrival time to a station
32	Blade batch type #
33	Size of batch being processed

Stations	Description
M	Station number (as assigned by SIMAN)

M = 1
to
M = 29
	Machine stations 1-29 respectively

30	Exit station for batches leaving the system

Sequences	Description
NS	Number sequence -- used for station routings

NS = 1
to
NS = 7
	Routings for blade types 1-7 respectively

File #	Description
1-29	Queue for batches waiting for machines prior to processing at stations 1-29 respectively (input queue--uses station # to determine queue #)

Tallies	Description
1-29	Wait times for machines at stations 1-29 respectively (wait in input queues)

DSTAT	Description

NQ(1)
to
NQ(29)
	Number in station input queues 1-29 respectively

NR(1)
to
NR(29)
	Resource utilization at stations 1-29 respectively

Parameters	Description
1 - 7	Blade interarrival times for blade types 1-7 respectively
8 - 21	Station service times for type 1 blades
22 - 35	Station service times for type 2 blades
36 - 49	Station service times for type 3 blades
50 - 62	Station service times for type 4 blades
63 - 75	Station service times for type 5 blades
76 - 93	Station service times for type 6 blades
94 -101	Station service times for type 7 blades

```
BEGIN:
;
;
;
;                          FILENAME: LEVEL1.MOD
;                         ======================
;
;
;
;  Conditions: Instantaneous Transfer of Materials
;              Fixed Routes
;              Uniform Scheduling Based On Yearly Induction Quantities
;              Fixed Batch Sizes
;              Fixed # Workstations at Each Workcenter
;              Exponential Service Distributions
;              No Breakdowns
;              Infinite Queue Sizes
;              FIFO Service Priority
;             *One Station Block Models All Stations
;             *In-line Commenting
;             *Machining Times Are For Batches, Not Blades
;
;---------------------------------------------------------------------
;
;       ***Assign Numeric Values to Batch Size Variables***
;
SYNONYMS: BS1 =  92:   BS2 = 100:   BS3 =  50:   BS4 =  90:
          BS5 =  94:   BS6 = 100:   BS7 =  50;
;
;---------------------------------------------------------------------
;
;           ***Create Arrivals of Batches to the System***
;
        CREATE,,4176.:EX(1,1):MARK(31);      Create Batch 1 Arrivals
        ASSIGN:A(32)=1;                      Batch Type # = A(32)
        ASSIGN:A(33)= 'BS1';                 Batch Size = A(33)
        ASSIGN:NS=1:NEXT(LOOP);              Assign Route to Batch 1
;
        CREATE,,4176:EX(2,1):MARK(31);       Create Batch 2 Arrivals
        ASSIGN:A(32)=2;                      Batch Type # = A(32)
        ASSIGN:A(33)= 'BS2';                 Batch Size = A(33)
        ASSIGN:NS=2:NEXT(LOOP);              Assign Route to Batch 2
;
        CREATE,,4176:EX(3,1):MARK(31);       Create Batch 3 Arrivals
        ASSIGN:A(32)=3;                      Batch Type # = A(32)
        ASSIGN:A(33) = 'BS3';                Batch Size = A(33)
        ASSIGN:NS=3:NEXT(LOOP);              Assign Route to Batch 3
;
        CREATE,,1392:EX(4,1):MARK(31);       Create Batch 4 Arrivals
        ASSIGN:A(32)=4;                      Batch Type # = A(32)
        ASSIGN:A(33) = 'BS4';                Batch Size = A(33)
        ASSIGN:NS=4:NEXT(LOOP);              Assign Route to Batch 4
;
        CREATE,,2088:EX(5,1):MARK(31);       Create Batch 5 Arrivals
        ASSIGN:A(32)=5;                      Batch Type # = A(32)
        ASSIGN:A(33) = 'BS5';                Batch Size = A(33)
        ASSIGN:NS=5:NEXT(LOOP);              Assign Route to Batch 5
;
        CREATE,,418:EX(6,1):MARK(31);        Create Batch 6 Arrivals
        ASSIGN:A(32)=6;                      Batch Type # = A(32)
```

```
         ASSIGN:A(33) = 'BS6';                Batch Size = A(33)
         ASSIGN:NS=6:NEXT(LOOP);              Assign Route to Batch 6
;
         CREATE,,348:EX(7,1):MARK(31);        Create Batch 7 Arrivals
         ASSIGN:A(32)=7;                      Batch Type # = A(32)
         ASSIGN:A(33) = 'BS7';                Batch Size = A(33)
         ASSIGN:NS=7:NEXT(LOOP);              Assign Route to Batch 7
;
;----------------------------------------------------------------------
;
;                   ***Model Station Visits***
;
LOOP     ROUTE:0.0,SEQ;
;
         STATION,1-29;                      When batches arrive to
station
         ASSIGN:A(M+1)=TNOW;                Mark arrival time in A(M+1)
         QUEUE,M;                           Place batch in station queue
         SEIZE:MACHINE(M);                  Seize machine when available
         TALLY:M,TNOW - A(M+1);             Record wait time in queue
         DELAY:A(1);                        Process the batch
         RELEASE:MACHINE(M):NEXT(LOOP);     Release machine; Go to next
;
;----------------------------------------------------------------------
;
;                 ***Dispose of Completed Batches***
;
         STATION,30;                   Record total time in system and
         TALLY:30,INT(31):DISPOSE;     Dispose of batch entitiy
END;
```

```
BEGIN;
;
;
;                         FILENAME: LEVEL1.EXP
;
;
;
PROJECT,NADEP,WORSHAM,10/03/88;
;
DISCRETE,300,35,30,30;
;
RESOURCES:1-29,MACHINE,1,1,1,8,1,1,1,1,2,10,
                        2,7,2,2,2,5,4,1,1, 2,
                        1,2,2,1,1,1,1,1;
;
SEQUENCES:
;
1,15,EX(8,1) / 6,EX(9,1) /10,EX(10,1)/24,EX(11,1)/18,EX(12,1)/
   4,EX(13,1)/12,EX(14,1)/11,EX(15,1)/10,EX(16,1)/ 6,EX(17,1)/
  18,EX(18,1)/16,EX(19,1)/24,EX(20,1)/ 6,EX(21,1)/30:
2,19,EX(22,1)/15,EX(23,1)/24,EX(24,1)/ 6,EX(25,1)/16,EX(26,1)/
  24,EX(27,1)/18,EX(28,1)/ 4,EX(29,1)/20,EX(30,1)/12,EX(31,1)/
  10,EX(32,1)/ 6,EX(33,1)/10,EX(34,1)/10,EX(35,1)/30:
3,15,EX(36,1)/15,EX(37,1)/15,EX(38,1)/ 6,EX(39,1)/20,EX(40,1)/
  15,EX(41,1)/15,EX(42,1)/ 4,EX(43,1)/16,EX(44,1)/20,EX(45,1)/
  10,EX(46,1)/10,EX(47,1)/ 6,EX(48,1)/24,EX(49,1)/30:
4, 6,EX(50,1)/10,EX(51,1)/24,EX(52,1)/18,EX(53,1)/ 4,EX(54,1)/
  12,EX(55,1)/11,EX(56,1)/10,EX(57,1)/ 6,EX(58,1)/18,EX(59,1)/
  16,EX(60,1)/24,EX(61,1)/ 6,EX(62,1)/30:
5, 6,EX(63,1)/10,EX(64,1)/24,EX(65,1)/18,EX(66,1)/ 4,EX(67,1)/
  12,EX(68,1)/11,EX(69,1)/10,EX(70,1)/ 6,EX(71,1)/18,EX(72,1)/
  16,EX(73,1)/24,EX(74,1)/ 6,EX(75,1)/30:
6,18,EX(76,1)/24,EX(77,1)/15,EX(78,1)/15,EX(79,1)/18,EX(80,1)/
  10,EX(81,1)/18,EX(82,1)/15,EX(83,1)/24,EX(84,1)/16,EX(85,1)/
  24,EX(86,1)/15,EX(87,1)/ 4,EX(88,1)/16,EX(89,1)/ 6,EX(90,1)/
  10,EX(91,1)/ 6,EX(92,1)/24,EX(93,1)/30:
7,18.EX(94,1)/22,EX(95,1)/10,EX(96,1)/15,EX(97,1)/24,EX(98,1)/
   9,EX(99,1)/ 5,EX(100,1)/5,EX(101,1)/30;
;
DSTAT:
;
       1,NQ(4).NO. IN QUE 4:             2,NQ(5),NO. IN QUE 5:
       3,NQ(6),NO. IN QUE 6:             4,NQ(9),NO. IN QUE 9:
       5,NQ(10),NO. IN QUE 10:           6,NQ(11),NO. IN QUE 11:
       7,NQ(12),NO  IN QUE 12:           8,NQ(15),NO. IN QUE 15:
       9,NQ(16),NO. IN QUE 16:          10,NQ(18),NO. IN QUE 18:
      11,NQ(20),NO. IN QUE 20:          12,NQ(22),NO. IN QUE 22:
      13,NQ(24),NO. IN QUE 24:          14,NR(4),UTIL. OF MACH 4:
      15,NR(5),UTIL. OF MACH 5:         16,NR(6),UTIL. OF MACH 6:
      17,NR(9),UTIL. OF MACH 9:         18,NR(10),UTIL. OF MACH 10:
      19,NR(11),UTIL. OF MACH 11:       20,NR(12),UTIL. OF MACH 12:
      21,NR(15),UTIL. OF MACH 15:       22,NR(16),UTIL. OF MACH 16:
      23,NR(18),UTIL. OF MACH 18:       24,NR(20),UTIL. OF MACH 20:
      25,NR(22),UTIL. OF MACH 22:       26,NR(24),UTIL. OF MACH 24;
;
PARAMETERS:
;
 1, 17.0:  2, 24.8:  3, 12.5:  4, 14.4:
 5, 19.6:  6, 19.2:  7, 11.7:
! TYPE 1 BLADE SERVICE TIMES
 8, 0.50:  9, 0.27: 10, 4.00: 11, 0.50: 12, 0.21: 13, 8.50: 14, 8.00:
```

```
15, 5.00: 16, 8.00: 17, 0.27: 18, 0.21: 19, 0.50: 20, 0.50: 21, 0.27:
! TYPE 2 BLADE SERVICE TIMES
22, 0.50: 23, 2.00: 24, 1.00: 25, 1.00: 26, 1.00: 27, 1.50: 28, 0.50:
29, 8.00: 30, 8.00: 31, 8.00: 32,10.00: 33, 1.00: 34, 1.50: 35, 2.50:
! TYPE 3 BLADE SERVICE TIMES
36, 0.50: 37, 1.50: 38, 2.50: 39, 1.00: 40, 8.00: 41, 1.50: 42, 5.00:
43, 8.00: 44, 1.00: 45,16.00: 46,10.00: 47, 2.50: 48, 1.00: 49, 1.50:
! TYPE 4 BLADE SERVICE TIMES
50, 0.27: 51, 4.00: 52, 0.50: 53, 0.21: 54, 8.50: 55, 8.00: 56, 5.00:
57, 8.00: 58, 0.27: 59, 0.21: 60, 0.50: 61, 0.50: 62, 0.27:
! TYPE 5 BLADE SERVICE TIMES
63, 0.27: 64, 4.00: 65, 0.50: 66, 0.21: 67, 8.50: 68, 8.00: 69, 5.00:
70, 8.00: 71, 0.27: 72, 0.21: 73, 0.50: 74, 0.50: 75, 0.27:
! TYPE 6 BLADE SERVICE TIMES
76, 0.50: 77, 1.50: 78, 2.50: 79, 1.00: 80, 0.50: 81, 8.00: 82, 0.50:
83, 2.50: 84, 1.50: 85, 1.50: 86, 1.50: 87, 2.50:88, 10.00: 89, 1.50:
90, 1.00:91, 16.00: 92, 1.00: 93, 1.50:
! TYPE 7 BLADE SERVICE TIMES
94, 0.25: 95, 8.00: 96, 6.25: 97, 5.00: 98, 1.00: 99, 4.00: 100, 4.00:
101, 1.50;
;
TALLIES: 1,TIME IN QUE 1:          2,TIME IN QUE 2:
         3,TIME IN QUE 3:          4,TIME IN QUE 4:
         5,TIME IN QUE 5:          6,TIME IN QUE 6:
         7,TIME IN QUE 7:          8,TIME IN QUE 8:
         9,TIME IN QUE 9:         10,TIME IN QUE 10:
        11,TIME IN QUE 11:        12,TIME IN QUE 12:
        13,TIME IN QUE 13:        14,TIME IN QUE 14:
        15,TIME IN QUE 15:        16,TIME IN QUE 16:
        17,TIME IN QUE 17:        18,TIME IN QUE 18:
        19,TIME IN QUE 19:        20,TIME IN QUE 20:
        21,TIME IN QUE 21:        22,TIME IN QUE 22:
        23,TIME IN QUE 23:        24,TIME IN QUE 24:
        25,TIME IN QUE 25:        26,TIME IN QUE 26:
        27,TIME IN QUE 27:        28,TIME IN QUE 28:
        29,TIME IN QUE 29:        30,TIME IN SYSTEM;
;
REPLICATE,1,0,2000;
END;
```

```
:----------------------------------------------------------------------:
:                                                                      :
:                                                                      :
:                        SIMAN Run Processor                           :
:                            Version 3.5                               :
:                      License Number 8810399                          :
:                                                                      :
:----------------------------------------------------------------------:
:                                                                      :
:   Systems Modeling Corporation licenses this program for use by :    :
:                                                                      :
:                        North Carolina State                          :
:                                                                      :
:   This program may only be used or copied according to the terms     :
:   of that license.                                                   :
:                                                                      :
:----------------------------------------------------------------------:


Please press <return> to begin the simulation.



          Recalling the   PROGRAM   file  c:\siman\level1\level1.p


SIMAN Run Processor Version 3.5
Copyright 1985, 1986, 1987 by Systems Modeling Corp.



Beginning execution of run number    1
```

```
                              SIMAN Summary Report

                              Run Number    1 of   1

Project:   NADEP
Analyst.   WORSHAM
Date   :   10/ 3/1988

Run ended at time      .2000E+04


                                Tally Variables
                                ---------------

Number Identifier           Average     Standard    Minimum     Maximum      Number
                                        Deviation   Value       Value        of Obs
------------------------------------------------------------------------------------

     1  TIME IN QUE 1         .00000      .00000      .00000      .00000           0
     2  TIME IN QUE 2         .00000      .00000      .00000      .00000           0
     3  TIME IN QUE 3         .00000      .00000      .00000      .00000           0
     4  TIME IN QUE 4         .00000      .00000      .00000      .00000         115
     5  TIME IN QUE 5        2.47148     4.15271      .00000    24.29517         226
     6  TIME IN QUE 6         .07056      .30872      .00000     3.19604         254
     7  TIME IN QUE 7         .00000      .00000      .00000      .00000           0
     8  TIME IN QUE 8         .00000      .00000      .00000      .00000           0
     9  TIME IN QUE 9         .04681      .32412      .00000     3.06934         115
    10  TIME IN QUE 10        .00000      .00000      .00000      .00000         341
    11  TIME IN QUE 11        .00000      .00000      .00000      .00000          31
    12  TIME IN QUE 12        .00000      .00000      .00000      .00000          32
    13  TIME IN QUE 13        .00000      .00000      .00000      .00000           0
    14  TIME IN QUE 14        .00000      .00000      .00000      .00000           0
    15  TIME IN QUE 15        .74164     1.94490      .00000    13.68176         435
    16  TIME IN QUE 16        .00000      .00000      .00000      .00000         191
    17  TIME IN QUE 17        .00000      .00000      .00000      .00000           0
    18  TIME IN QUE 18        .05659      .25510      .00000     3.06378         425
    19  TIME IN QUE 19        .00000      .00000      .00000      .00000           0
    20  TIME IN QUE 20        .00000      .00000      .00000      .00000           0
    21  TIME IN QUE 21        .00000      .00000      .00000      .00000           0
    22  TIME IN QUE 22        .63278     2.15606      .00000    12.85120         117
    23  TIME IN QUE 23        .00000      .00000      .00000      .00000           0
    24  TIME IN QUE 24        .90119     2.19214      .00000    17.62292         501
    25  TIME IN QUE 25        .00000      .00000      .00000      .00000           0
    26  TIME IN QUE 26        .00000      .00000      .00000      .00000           0
    27  TIME IN QUE 27        .00000      .00000      .00000      .00000           0
    28  TIME IN QUE 28        .00000      .00000      .00000      .00000           0
    29  TIME IN QUE 29        .00000      .00000      .00000      .00000           0
    30  TIME IN SYSTEM      46.62276    23.25068    12.16431   129.44310         221
```

```
                               Discrete Change Variables
                               -------------------------

Number Identifier            Average      Standard     Minimum      Maximum      Time
                                          Deviation    Value        Value        Period
-------------------------------------------------------------------------------------------

   1   NO. IN QUE 4             .00          .00          .00          .00       2000.00
   2   NO. IN QUE 5             .28          .68          .00         4.00       2000.00
   3   NO. IN QUE 6             .01          .10          .00         2.00       2000.00
   4   NO. IN QUE 9             .00          .05          .00         1.00       2000.00
   5   NO. IN QUE 10            .00          .00          .00          .00       2000.00
   6   NO. IN QUE 11            .00          .00          .00          .00       2000.00
   7   NO. IN QUE 12            .00          .00          .00          .00       2000.00
   8   NO. IN QUE 15            .16          .62          .00         5.00       2000.00
   9   NO. IN QUE 16            .00          .00          .00          .00       2000.00
  10   NO. IN QUE 18            .01          .12          .00         2.00       2000.00
  11   NO. IN QUE 20            .00          .00          .00          .00       2000.00
  12   NO. IN QUE 22            .04          .23          .00         2.00       2000.00
  13   NO. IN QUE 24            .23          .77          .00         6.00       2000.00
  14   UTIL. OF MACH 4          .59          .79          .00         4.00       2000.00
  15   UTIL. OF MACH 5          .33          .47          .00         1.00       2000.00
  16   UTIL. OF MACH 6          .10          .29          .00         1.00       2000.00
  17   UTIL. OF MACH 9          .23          .47          .00         2.00       2000.00
  18   UTIL. OF MACH 10        1.49         1.48          .00         8.00       2000.00
  19   UTIL. OF MACH 11         .08          .29          .00         2.00       2000.00
  20   UTIL. OF MACH 12         .14          .47          .00         3.00       2000.00
  21   UTIL. OF MACH 15         .59          .76          .00         2.00       2000.00
  22   UTIL. OF MACH 16         .14          .38          .00         3.00       2000.00
  23   UTIL. OF MACH 18         .07          .26          .00         1.00       2000.00
  24   UTIL. OF MACH 20         .00          .00          .00          .00       2000.00
  25   UTIL. OF MACH 22         .46          .67          .00         2.00       2000.00
  26   UTIL. OF MACH 24         .30          .46          .00         1.00       2000.00

Run Time :  1 Minute(s) and 37 Second(s)


Stop - Program terminated.
```

Attachment 2

Sample Level 2 Simulation Files

SIMAN Variable Dictionary: Level2

Attribute #	Description
1	Station batch service time -- updated automatically for each station in sequence
2	Machine attrition rating
3	Initial (entrance) batch size
4	Index # of AGV being used for transport to next station
5	Batch arrival time to system
6	Tally variable for individual batch flow times
7	Wait time in queue attribute
8	Dynamic (attrited) batch size

Stations	Description
M	Station number (as assigned by SIMAN)

M = 1
to
M = 29
	Machining stations 1-29 respectively

30	Exit station for batches leaving the system
31	Entrance station for batches arriving to the system

Sequences	Description
NS	Number sequence -- used for station routings

NS = 1
to
NS = 7
	Routings for blade types 1-7 respectively

File #	Description
1-29	Queue for batches waiting for machines prior to processing at stations 1-29 respectively (input queue)
30	Queue for new batches waiting at entrance station for transport to first station in sequence
31-59	Queue for batches waiting for AGV transport to next station in sequence at stations 1-29 respectively (output queue)

Global Var	Description
X(1)	Number of batches in system
X(2)	Number of batches through system
X(3)	Number of blades through system

Tallies	Description
1-29	Wait times for machines at stations 1-29 respectively (wait in input queues)
30-58	Wait times for AGVs at stations 1-29 respectively (wait in output queues)
59	Batch type 1 time in system
60	Batch type 2 time in system
61	Batch type 3 time in system
62	Batch type 4 time in system
63	Batch type 5 time in system
64	Batch type 6 time in system
65	Batch type 7 time in system
66	Overall average batch flowtime
67	Wait time for repair

DSTAT	Description

NQ(1)
to
NQ(29)
	Number in station input (Mach) queues 1-29 respectively

NQ(30)	Number in entrance station (AGV) queue

NQ(31)
to
NQ(59)
	Number in station output (AGV) queues 1-29 respectively

NQ(60)	Number in repair queue

NR(1)
to
NR(29)
	Resource utilization at stations 1-29 respectively

NR(30)	Repairmen resources utilization

MR(1)
to
MR(29)
	Number of machines up (not broken down) at stations 1-29 respectively

NT(1)	Number of busy transporters (AGVs) (transporting)
MT(1)	Number of transporters (AGVs) up (not broken down)
X(1)	Number of batches in system
X(2)	Number of batches through system
X(3)	Number of blades through system

Parameters	Description
1 - 7	Blade interarrival times for blade types 1-7 respectively
8 - 21	Station service times for type 1 blades
22 - 35	Station service times for type 2 blades
36 - 49	Station service times for type 3 blades
50 - 62	Station service times for type 4 blades
63 - 75	Station service times for type 5 blades
76 - 93	Station service times for type 6 blades
94 -101	Station service times for type 7 blades

Parameters	Description
102	AGV failure rate parameter
103	AGV repair rate parameter
104	Machine (common) failure rate parameter
105	Machine (common) repair rate parameter

```
BEGIN;
;
;                              FILENAME: LEVEL2.MOD
;                              ====================
;
;-------------------------------------------------------------------------
;
; Simulation Conditions:
;
;      Transportation System Included
;      Fixed Routes
;      Uniform Scheduling Based On Yearly Induction Quantities
;      Variable Batch Sizes to Account for Attrition
;      Fixed # Permanent Workstations at Each Workcenter
;      Exponential Service Distributions
;      AGV Breakdowns Modeled
;      Shortest Distance Priority On AGV's
;      Machine Breakdowns Modelled With Alter Block (.mod)
;      Exponential Breakdowns and Repairs
;      Infinite Queue Sizes
;      FIFO Service Priority
;      One Station Block Models All Stations
;      Output Queue Tallies Included
;      Collect Statistics on Number of Batches in System
;      Reduced Number of Attributes Per Entity
;      Individual Flowtime Statistics Collected
;      Variable Offset Times For Batch Creations
;      Capacity Limit on Number of Repairmen (10)
;      Collect Statistics On # Batches & Blades Through System
;
;-------------------------------------------------------------------------
;
;                         *** Attribute Dictionary ***
;
;
;  M          - Station Number
;  NS         - Sequence Number / Blade Type (for routing to stations)
;  A(1)       - Batch Machining Time
;  A(2)       - Machine Attrition Rating
;  A(3)       - Initial (Entrance) Batch Size
;  A(4)       - AGV Number
;  A(5)       - Batch Arrival Time to System
;  A(6)       - Tally Variable for Individual Flowtimes
;  A(7)       - Wait Time Attribute
;  A(8)       - Variable (Attrited) Batch Size
;  A(9)       - Animation (Entity) Variable
;
;-------------------------------------------------------------------------
;
;              *** Create Arrivals of Batches to the System ***
;
;
        CREATE,,EX(1,2):EX(1,1):MARK(5);       Create batch 1 arrivals
        ASSIGN:A(3) = 92;                      Batch size = A(3)
        ASSIGN:NS=1:NEXT(BEG);                 Assign route to batch 1
;
        CREATE,,EX(2,2):EX(2,1):MARK(5);       Create batch 2 arrivals
        ASSIGN:A(3) = 100;                     Batch size = A(3)
        ASSIGN:NS=2:NEXT(BEG);                 Assign route to batch 2
;
```

```
        CREATE,,EX(3,2):EX(3,1):MARK(5);     Create batch 3 arrivals
        ASSIGN:A(3) = 50;                    Batch size = A(3)
        ASSIGN:NS=3:NEXT(BEG);               Assign route to batch 3
;
        CREATE,,EX(4,2):EX(4,1):MARK(5);     Create batch 4 arrivals
        ASSIGN:A(3) = 90;                    Batch size = A(3)
        ASSIGN:NS=4:NEXT(BEG);               Assign route to batch 4
;
        CREATE,,EX(5,2):EX(5,1):MARK(5);     Create batch 5 arrivals
        ASSIGN:A(3) = 94;                    Batch size = A(3)
        ASSIGN:NS=5:NEXT(BEG);               Assign route to batch 5
;
        CREATE,,EX(6,2):EX(6,1):MARK(5);     Create batch 6 arrivals
        ASSIGN:A(3) = 100;                   Batch size = A(3)
        ASSIGN:NS=6:NEXT(BEG);               Assign route to batch 6
;
        CREATE,,EX(7,2):EX(7,1):MARK(5);     Create batch 7 arrivals
        ASSIGN:A(3) = 50;                    Batch size = A(3)
        ASSIGN:NS=7:NEXT(BEG);               Assign route to batch 7
;
;--------------------------------------------------------------------
;
;                ********** Before Routing **********
;
;
BEG     ASSIGN:X(1)=X(1)+1;   Increment total # batches in system
        ASSIGN:A(9)=NS;       Assign batch type to animation variable
        ASSIGN:A(8)=A(3);     Assign variable batch size initial value
        ROUTE:0.0,31;         Send batch to entrance station
;
;--------------------------------------------------------------------
;
;                ********** Entrance Station **********
;
        STATION,31;                    When batches arrive to system
        ASSIGN:A(7)=TNOW;              Mark arrival time to station
        QUEUE,30;                      Q batches waiting for transport
        REQUEST:AGV(SDS,4);            Request AGV--A(4)=AGV#
        TALLY:59,TNOW - A(7);          Record wait time in output queue
        TRANSPORT:AGV(A(4)),SEQ;       Move to 1st station w/AGV # A(4)
;
;--------------------------------------------------------------------
;
;           ********** Model Resource Station Visits **********
;
;
        STATION,1-29;               When batches arrive to a station
        FREE:AGV(A(4));             Free the AGV that transported batch
        ASSIGN:A(7)=TNOW;           Mark arrival time in A(7)
        QUEUE,M;                    Place batch in station input queue
        SEIZE:MACHINE(M);           Seize machine when available
        TALLY:M,TNOW - A(7);        Record wait time in input queue
        DELAY:A(1)/A(3)*A(8);         Process the batch
        ASSIGN:A(8)=A(8)*A(2);        Decrement batch size (attrition)
        RELEASE:MACHINE(M);           Release machine
        ASSIGN:A(7)=TNOW;             Mark mach completion time in A(7)
        QUEUE(M+30);                   Place batch in station output queue
        REQUEST:AGV(SDS,4);             Request AGV--A(4)=AGV#
        TALLY:M+29,TNOW - A(7);         Record wait time in output queue
        TRANSPORT:AGV(A(4)),SEQ;           Move via AGV to next station
```

```
;
;------------------------------------------------------------------------
;
;                       *** Exit Station ***
;
;
          STATION,30;              When batches arrive to the exit station:
          FREE:AGV(A(4));          Free the AGV that transported batch
          ASSIGN:A(7)=NS+58;       Assign A(7)=Batch #+59--Use in next line
          TALLY:A(7),INT(5);         Record individual batch times in sys
          ASSIGN:X(1)=X(1)-1;          Decrement tot # batches in system
          ASSIGN:X(2)=X(2)+1;           Increment tot # batches thru system
          ASSIGN:X(3)=X(3)+A(8);        Increment tot # blades thru system
          TALLY:66,INT(5):DISPOSE;      Record batch summary times in sys
;
;------------------------------------------------------------------------
;
;                     *** Machine Breakdowns ***
;
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 1 bd entity
          ASSIGN:A(1)=1;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(1):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 2 bd entity
          ASSIGN:A(1)=2;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(2):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 3 bd entity
          ASSIGN:A(1)=3;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(3):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/8;     Create station 4 bd entity
          ASSIGN:A(1)=4;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(4):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 5 bd entity
          ASSIGN:A(1)=5;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(5):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 6 bd entity
          ASSIGN:A(1)=6;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(6):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 7 bd entity
          ASSIGN:A(1)=7;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(7):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 8 bd entity
          ASSIGN:A(1)=8;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(8):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/2;     Create station 9 bd entity
          ASSIGN:A(1)=9;                     Assign station # to A(1)
          ASSIGN:A(2)=MR(9):NEXT(DECIDE);    A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/10;    Create station 10 bd entity
          ASSIGN:A(1)=10;                    Assign station # to A(1)
          ASSIGN:A(2)=MR(10):NEXT(DECIDE);   A(2)=# up, go to decision
;
```

```
     CREATE,,EX(104,2):EX(104,1)/2;       Create station 11 bd entity
     ASSIGN:A(1)=11;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(11):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/7;       Create station 12 bd entity
     ASSIGN:A(1)=12;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(12):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/2;       Create station 13 bd entity
     ASSIGN:A(1)=13;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(13):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/2;       Create station 14 bd entity
     ASSIGN:A(1)=14;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(14):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/2;       Create station 15 bd entity
     ASSIGN:A(1)=15;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(15):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/5;       Create station 16 bd entity
     ASSIGN:A(1)=16;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(16):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/4;       Create station 17 bd entity
     ASSIGN:A(1)=17;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(17):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/1;       Create station 18 bd entity
     ASSIGN:A(1)=18;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(18):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/1;       Create station 19 bd entity
     ASSIGN:A(1)=19;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(19):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/2;       Create station 20 bd entity
     ASSIGN:A(1)=20;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(20):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/1;       Create station 21 bd entity
     ASSIGN:A(1)=21;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(21):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/2;       Create station 22 bd entity
     ASSIGN: A(1)=22;                     Assign station # to A(1)
     ASSIGN:A(2)=MR(22):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/2;       Create station 23 bd entity
     ASSIGN:A(1)=23;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(23):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/1;       Create station 24 bd entity
     ASSIGN:A(1)=24;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(24):NEXT(DECIDE);     A(2)=# up, go to decision
;
     CREATE,,EX(104,2):EX(104,1)/1;       Create station 25 bd entity
     ASSIGN:A(1)=25;                      Assign station # to A(1)
     ASSIGN:A(2)=MR(25):NEXT(DECIDE);     A(2)=# up, go to decision
;
```

```
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 26 bd entity
          ASSIGN:A(1)=26;                    Assign station # to A(1)
          ASSIGN:A(2)=MR(26):NEXT(DECIDE);   A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 27 bd entity
          ASSIGN:A(1)=27;                    Assign station # to A(1)
          ASSIGN:A(2)=MR(27):NEXT(DECIDE);   A(2)=# up, go to decision
;
          CREATE,,EX(104,2):EX(104,1)/1;     Create station 28 bd entity
          ASSIGN:A(1)=28;                    Assign station # to A(1)
          ASSIGN:A(2)=MR(28):NEXT(DECIDE);   A(2)=# up; go to decision
;
;
DECIDE    BRANCH,1:                          !If all machines down,
             IF,A(2).EQ.0,DISP:              !Dispose of breakdown ent
             ELSE,MACHBD;                    Otherwise, perf bd/repair
;
;
MACHBD    ALTER:MACHINE(A(1)),-1;            Decrement # machines by 1
          ASSIGN:A(3)=TNOW;                  Record entrance to rep Q
          QUEUE,60;                          Wait in queue 60 for repman
          SEIZE:REPMAN;                      Seize repairman when avail
          TALLY:67,TNOW-A(3);                Record wait time in rep Q
          DELAY:EX(105,1);                   Repair the machine
          RELEASE:REPMAN;                    Free repairman when comp
          ALTER:MACHINE(A(1)),1:DISPOSE;     Increment # machines by 1
;                                            when repair is complete
;
;
;-------------------------------------------------------------------
;
;
;                           *** AGV Breakdowns ***
;
;
;
          CREATE,,EX(102,2):EX(102,1);     Create AGV 1 Breakdown
          ASSIGN:A(4)=1;
          BRANCH,1:                        !If AGV Already Down, Dispose
             IF,IT(1,1).EQ.2,DISP:         !of Breakdown Entity
             ELSE, AGVBD;                  Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);     Create AGV 2 Breakdown
          ASSIGN:A(4)=2;
          BRANCH,1:                        !If AGV Already Down, Dispose
             IF,IT(1,2).EQ.2,DISP:         !of Breakdown Entity
             ELSE, AGVBD;                  Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);     Create AGV 3 Breakdown
          ASSIGN:A(4)=3;
          BRANCH,1:                        !If AGV Already Down, Dispose
             IF,IT(1,3).EQ.2,DISP:         !of Breakdown Entity
             ELSE, AGVBD;                  Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);     Create AGV 4 Breakdown
          ASSIGN:A(4)=4;
          BRANCH,1:                        !If AGV Already Down, Dispose
             IF,IT(1,4).EQ.2,DISP:         !of Breakdown Entity
             ELSE, AGVBD;                  Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);     Create AGV 5 Breakdown
          ASSIGN:A(4)=5;
          BRANCH,1:                        !If AGV Already Down, Dispose
```

```
              IF,IT(1,5).EQ.2,DISP:          !of Breakdown Entity
              ELSE, AGVBD;                   Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);       Create AGV 6 Breakdown
          ASSIGN:A(4)=6;
          BRANCH,1:                          !If AGV Already Down, Dispose
              IF,IT(1,6).EQ.2,DISP:          !of Breakdown Entity
              ELSE, AGVBD;                   Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);       Create AGV 7 Breakdown
          ASSIGN:A(4)=7;
          BRANCH,1:                          !If AGV Already Down, Dispose
              IF,IT(1,7).EQ.2,DISP:          !of Breakdown Entity
              ELSE, AGVBD;                   Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);       Create AGV 8 Breakdown
          ASSIGN:A(4)=8;
          BRANCH,1:                          !If AGV Already Down, Dispose
              IF,IT(1,8).EQ.2,DISP:          !of Breakdown Entity
              ELSE, AGVBD;                   Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);       Create AGV 9 Breakdown
          ASSIGN:A(4)=9;
          BRANCH,1:                          !If AGV Already Down, Dispose
              IF,IT(1,9).EQ.2,DISP:          !of Breakdown Entity
              ELSE, AGVBD;                   Otherwise, Perform Breakdown
;
          CREATE,,EX(102,2):EX(102,1);       Create AGV 10 Breakdown
          ASSIGN:A(4)=10;
          BRANCH,1:                          !If AGV Already Down, Dispose
              IF,IT(1,10).EQ.2,DISP:         !of Breakdown Entity
              ELSE, AGVBD;                   Otherwise, Perform Breakdown
;
AGVBD     HALT:AGV(A(4));                    Stop AGV for Repair
          DELAY:EX(103,1);                   Repair AGV
          ACTIVATE:AGV(A(4)):DISPOSE;        Re-activate AGV After Repair
;
DISP      DELAY:0:DISPOSE;
;---------------------------------------------------------------------------
;
END;
```

```
BEGIN:
;
;
;                          FILENAME: LEVEL2.EXP
;
;
PROJECT,NADEP,WORSHAM,7/12/88;
;
;Allow 650 entities, 10 attributes, 60 files (queues), and 31 stations
;
DISCRETE,650.10,60,31,9;
;
RESOURCES:1-29,MACHINE,1,1,1,8,1,1,1,1,3,10,
                          8,8,2,2,2,5,4,1,1, 2,
                          1,4,4,1,1,1,1,1:
             30,REPMAN,10;
;
Transporters: 1,AGV,10,1,1000.0, 4-A, 5-A, 6-A, 9-A,10-A,
                                 12-A,15-A,16-A,18-A,31-A;
;
Distances:1,1-31,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 1
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 2
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 3
11,28,1,1,7,20,13,8,1,1,32,4,1,31,1,16,1,32,20,1,19,1,1,1,1,31,32/! from 4
20,1,1,16,30,23,18,1,1,30,8,1,25,1,26,1,24,30,11,1,1,1,1,1,23,24/! from 5
1,1,22,19,26,31,1,1,13,25,1,8,1,23,1,7,19,12,1,1,1,1,1,3,4/! from 6
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 7
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 8
14,7,2,1,1,26,10,1,25,1,10,1,26,14,25,1,1,1,1,1,35,36/! from 9
8,13,1,1,13,23,1,12,1,5,1,13,2,28,1,1,1,1,1,22,23/! from 10
6,1,1.20,16,1,19,1,4,1,20,8,31.1,1,1,1,1,29,30/! from 11
1,1,25,11,1.24,1,9,1,25,13,26,1,1,1,1,1,34,35/! from 12
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 13
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 14
36,1,7.1,18,1,8,13,23,1,1,1.1,1,16,17/! from 15
1,30,1,19,1,29,23,16,1,1,1,1,1,28.29/! from 16
1,1,1,1,1,1,1,1,1,1,1,1,1,1/! from 17
1,16.1,2,12,17,1,1,1,1,1,11,12/! from 18
1,1,1,1,1,1,1,1,1,1,1,1/! from 19
1,17,5,32,1,1,1,1,1,26,27/! from 20
1,1,1,1,1,1,1,1,1,1/! from 21
13,16,1,1,1,1,1,10,11/! from 22
28,1,1,1,1,1,22,23/! from 23
1,1,1,1,1,15,16/! from 24
1,1,1,1,1,1/! from 25
1,1,1,1,1/! from 26
1,1,1,1/! from 27
1,1,1/! from 28
1,1/! from 29
1;from 30
;
SEQUENCES:
1, 15,EX(  8,1), .95 /  6,EX(  9,1), .95 / 10,EX( 10,1), .95 /
   24,EX( 11,1), .95 / 18,EX( 12,1), .95 /  4,EX( 13,1), .95 /
   12,EX( 14,1), .95 / 11,EX( 15,1), .95 / 10,EX( 16,1), .95 /
    6,EX( 17,1), .95 / 18,EX( 18,1), .95 / 16,EX( 19,1), .95 /
   24,EX( 20,1), .95 /  6,EX( 21,1), .95 / 30:
2, 18,EX( 22,1), .95 / 15,EX( 23,1), .95 / 24,EX( 24,1), .95 /
    6,EX( 25,1), .95 / 16,EX( 26,1), .95 / 24,EX( 27,1), .95 /
    18,EX( 28,1),.95 /  4,EX( 29,1),.95 / 20,EX( 30,1),.95 /
    12,EX( 31,1),.95 / 10,EX( 32,1),.95 /  6,EX( 33,1),.95 /
    10,EX( 34,1),.95 / 10,EX( 35,1),.95 / 30;
 3. 15,EX( 36,1),.95 / 15,EX( 37,1),.95 / 15,EX( 38,1),.95 /
     6,EX( 39,1),.95 / 20,EX( 40,1),.95 / 15,EX( 41,1),.95 /
    15,EX( 42,1),.95 /  4,EX( 43,1),.95 / 16,EX( 44,1),.95 /
    20,EX( 45,1),.95 / 10,EX( 46,1),.95 / 10,EX( 47,1),.95 /
     6,EX( 48,1),.95 / 24,EX( 49,1),.95 / 30;
 4,  6,EX( 50,1),.95 / 10,EX( 51,1),.95 / 24,EX( 52,1),.95 /
    18,EX( 53,1),.95 /  4,EX( 54,1),.95 / 12,EX( 55,1),.95 /
    11,EX( 56,1),.95 / 10,EX( 57,1),.95 /  6,EX( 58,1),.95 /
    18,EX( 59,1),.95 / 16,EX( 60,1),.95 / 24,EX( 61,1),.95 /
     6,EX( 62,1),.95 / 30;
 5.  6,EX( 63,1),.95 / 10,EX( 64,1),.95 / 24,EX( 65,1),.95 /
    18,EX( 66,1),.95 /  4,EX( 67,1),.95 / 12,EX( 68,1),.95 /
    11,EX( 69,1),.95 / 10,EX( 70,1),.95 /  6,EX( 71,1),.95 /
    18,EX( 72,1),.95 / 16,EX( 73,1),.95 / 24,EX( 74,1),.95 /
     6,EX( 75,1),.95 / 30;
 6. 18,EX( 76,1),.95 / 24,EX( 77,1),.95 / 15,EX( 78,1),.95 /
    15,EX( 79,1),.95 / 18,EX( 80,1),.95 / 10,EX( 81,1),.95 /
    18,EX( 82,1),.95 / 15,EX( 83,1),.95 / 24,EX( 84,1),.95 /
    16,EX( 85,1),.95 / 24,EX( 86,1),.95 / 15,EX( 87,1),.95 /
     4,EX( 88,1),.95 / 16,EX( 89,1),.95 /  6,EX( 90,1),.95 /
    10,EX( 91,1),.95 /  6,EX( 92,1),.95 / 24,EX( 93,1),.95 /
    30;
 7. 18,EX( 94,1),.95 / 22,EX( 95,1),.95 / 10,EX( 96,1),.95 /
    15,EX( 97,1),.95 / 24,EX( 98,1),.95 /  9,EX( 99,1),.95 /
     5,EX(100,1),.95 /  5,EX(101,1),.95 / 30;
;
DSTAT;
          1,NQ(1) ,# IN STN  1 QUE:     2,NQ(2) ,# IN STN  2 QUE:
          3,NQ(3) ,# IN STN  3 QUE:     4,NQ(4) ,# IN STN  4 QUE:
          5,NQ(5) ,# IN STN  5 QUE:     6,NQ(6) ,# IN STN  6 QUE:
          7,NQ(7) ,# IN STN  7 QUE:     8,NQ(8) ,# IN STN  8 QUE:
          9,NQ(9) ,# IN STN  9 QUE:    10,NQ(10),# IN STN 10 QUE:
         11,NQ(11),# IN STN 11 QUE:    12,NQ(12),# IN STN 12 QUE:
         13,NQ(13),# IN STN 13 QUE:    14,NQ(14),# IN STN 14 QUE:
         15,NQ(15),# IN STN 15 QUE:    16,NQ(16),# IN STN 16 QUE:
         17,NQ(17),# IN STN 17 QUE:    18,NQ(18),# IN STN 18 QUE:
         19,NQ(19),# IN STN 19 QUE:    20,NQ(20),# IN STN 20 QUE:
         21,NQ(21),# IN STN 21 QUE:    22,NQ(22),# IN STN 22 QUE:
         23,NQ(23),# IN STN 23 QUE:    24,NQ(24),# IN STN 24 QUE:
         25,NQ(25),# IN STN 25 QUE:    26,NQ(26),# IN STN 26 QUE:
         27,NQ(27),# IN STN 27 QUE:    28,NQ(28),# IN STN 28 QUE:
         29,NQ(29),# IN STN 29 QUE:
         30,NR(1) ,UTIL OF STN  1:      31,NR(2) ,UTIL OF STN  2:
         32,NR(3) ,UTIL OF STN  3:      33,NR(4) ,UTIL OF STN  4:
         34,NR(5) ,UTIL OF STN  5:      35,NR(6) ,UTIL OF STN  6:
         36,NR(7) ,UTIL OF STN  7:      37,NR(8) ,UTIL OF STN  8:
         38,NR(9) ,UTIL OF STN  9:      39,NR(10),UTIL OF STN 10:
         40,NR(11),UTIL OF STN 11:      41,NR(12),UTIL OF STN 12:
         42,NR(13),UTIL OF STN 13:      43,NR(14),UTIL OF STN 14:
         44,NR(15),UTIL OF STN 15:      45,NR(16),UTIL OF STN 16:
         46,NR(17),UTIL OF STN 17:      47,NR(18),UTIL OF STN 18:
         48,NR(19),UTIL OF STN 19:      49,NR(20),UTIL OF STN 20:
         50,NR(21),UTIL OF STN 21:      51,NR(22),UTIL OF STN 22:
         52,NR(23),UTIL OF STN 23:      53,NR(24),UTIL OF STN 24:
         54,NR(25),UTIL OF STN 25:      55,NR(26),UTIL OF STN 26:
         56,NR(27),UTIL OF STN 27:      57,NR(28),UTIL OF STN 28:
         58,NR(29),UTIL OF STN 29:
```

```
     59,NQ(30),# IN BEG AGV QUE:
     60,NQ(31),# IN AGV QUE  @1:   61,NQ(32),# IN AGV QUE  @2:
     62,NQ(33),# IN AGV QUE  @3:   63,NQ(34),# IN AGV QUE  @4:
     64,NQ(35),# IN AGV QUE  @5:   65,NQ(36),# IN AGV QUE  @6.
     66,NQ(37),# IN AGV QUE  @7:   67,NQ(38),# IN AGV QUE  @8:
     68,NQ(39),# IN AGV QUE  @9:   69,NQ(40),# IN AGV QUE @10:
     70,NQ(41),# IN AGV QUE @11:   71,NQ(42),# IN AGV QUE @12.
     72,NQ(43),# IN AGV QUE @13:   73,NQ(44),# IN AGV QUE @14:
     74,NQ(45),# IN AGV QUE @15:   75,NQ(46),# IN AGV QUE @16:
     76,NQ(47),# IN AGV QUE @17:   77,NQ(48),# IN AGV QUE @18:
     78,NQ(49),# IN AGV QUE @19:   79,NQ(50),# IN AGV QUE @20:
     80,NQ(51),# IN AGV QUE @21:   81,NQ(52),# IN AGV QUE @22:
     82,NQ(53),# IN AGV QUE @23:   83,NQ(54),# IN AGV QUE @24:
     84,NQ(55),# IN AGV QUE @25:   85,NQ(56),# IN AGV QUE @26:
     86,NQ(57),# IN AGV QUE @27:   87,NQ(58),# IN AGV QUE @28:
     88,NQ(59),# IN AGV QUE @29:
     89,MT(1),# OF AGVs UP:        90,NT(1),# OF BUSY AGVs:
     91,MR(1),# OF MACH @1 UP:     92,MR(2),# OF MACH @2 UP:
     93,MR(3),# OF MACH @3 UP:     94,MR(4),# OF MACH @4 UP:
     95,MR(5),# OF MACH @5 UP:     96,MR(6),# OF MACH @6 UP:
     97,MR(7),# OF MACH @7 UP:     98,MR(8),# OF MACH @8 UP:
     99,MR(9),# OF MACH @9 UP:    100,MR(10),# OF MACH @10 UP:
    101,MR(11),# OF MACH @11 UP:  102,MR(12),# OF MACH @12 UP:
    103,MR(13),# OF MACH @13 UP:  104,MR(14),# OF MACH @14 UP:
    105,MR(15),# OF MACH @15 UP:  106,MR(16),#  F MACH @16 UP:
    107,MR(17),# OF MACH @17 UP:  108,MR(18),# OF MACH @18 UP:
    109,MR(19),# OF MACH @19 UP:  110,MR(20),# OF MACH @20 UP:
    111,MR(21),# OF MACH @21 UP:  112,MR(22),# OF MACH @22 UP:
    113,MR(23),# OF MACH @23 UP:  114,MR(24),# OF MACH @24 UP:
    115,MR(25),# OF MACH @25 UP:  116,MR(26),# OF MACH @26 UP:
    117,MR(27),# OF MACH @27 UP:  118,MR(28),# OF MACH @28 UP:
    119,MR(29),# OF MACH @29 UP:
    120,NQ(60),# IN REP QUEUE:    121,NR(30),UTIL OF REPMEN:
    122,X(1),BATCHES IN SYS:      123,X(2),BATCHES THRU SYS:
    124,X(3),BLADES THRU SYS;
;
parameters:
! interarrival times for the blades
 1, 17.0:  2, 24.8:  3, 12.5:  4, 14.4:
 5, 19.6:  6, 19.2:  7, 11.7:
! type 1 blade batch service times
 8, 0.50:  9, 0.27: 10, 4.00: 11, 0.50: 12, 0.21: 13, 8.50: 14, 8.00:
15, 5.00: 16, 8.00: 17, 0.27: 18, 0.21: 19, 0.50: 20, 0.50: 21, 0.27:
! type 2 blade batch service times
22, 0.50: 23, 2.00: 24, 1.00: 25, 1.00: 26, 1.00: 27, 1.50: 28, 0.50:
29, 8.00: 30, 8.00: 31, 8.00: 32,10.00: 33, 1.00: 34, 1.50: 35, 2.50:
! type 3 blade batch service times
36, 0.50: 37, 1.50: 38, 2.50: 39, 1.00: 40, 8.00: 41, 1.50: 42, 5.00:
43, 8.00: 44, 1.00: 45,16.00: 46,10.00: 47, 2.50: 48, 1.00: 49, 1.50:
! type 4 blade batch service times
50, 0.27: 51, 4.00: 52, 0.50: 53, 0.21: 54, 8.50: 55, 8.00: 56, 5.00:
57, 8.00: 58, 0.27: 59, 0.21: 60, 0.50: 61, 0.50: 62, 0.27:
! type 5 blade batch service times
63, 0.27: 64, 4.00: 65, 0.50: 66, 0.21: 67, 8.50: 68, 8.00: 69, 5.00:
70, 8.00: 71, 0.27: 72, 0.21: 73, 0.50: 74, 0.50: 75, 0.27:
! type 6 blade batch service times
76, 0.50: 77, 1.50: 78, 2.50: 79, 1.00: 80, 0.50: 81, 8.00: 82, 0.50:
83, 2.50: 84, 1.50: 85, 1.50: 86, 1.50: 87, 2.50:88, 10.00: 89, 1.50:
90, 1.00:91, 16.00: 92, 1.00: 93, 1.50:
! type 7 blade batch service times
```

```
94, 0.25: 95, 8.00: 96, 6.25: 97, 5.00: 98, 1.00: 99, 4.00: 100, 4.00:
101, 1.50:
! agv failure and repair times
102, 160: 103, 8:
! machine failure and repair times
104, 160: 105, 16;
;
tallies:1,wait for mach  1:          2,wait for mach  2:
        3,wait for mach  3:          4,wait for mach  4:
        5,wait for mach  5:          6,wait for mach  6:
        7,wait for mach  7:          8,wait for mach  8:
        9,wait for mach  9:         10,wait for mach 10:
       11,wait for mach 11:         12,wait for mach 12:
       13,wait for mach 13:         14,wait for mach 14:
       15,wait for mach 15:         16,wait for mach 16:
       17,wait for mach 17:         18,wait for mach 18:
       19,wait for mach 19:         20,wait for mach 20:
       21,wait for mach 21:         22,wait for mach 22:
       23,wait for mach 23:         24,wait for mach 24:
       25,wait for mach 25:         26,wait for mach 26:
       27,wait for mach 27:         28,wait for mach 28:
       29,wait for mach 29:         30,wait for agv  @1:
       31,wait for agv  @2:         32,wait for agv  @3:
       33,wait for agv  @4:         34,wait for agv  @5:
       35 wait for agv  @6:         36,wait for agv  @7:
       37,wait for agv  @8:         38,wait for agv  @9:
       39,wait for agv @10:         40,wait for agv @11:
       41,wait for agv @12:         42,wait for agv @13:
       43,wait for agv @14:         44,wait for agv @15:
       45,wait for agv @16:         46,wait for agv @17:
       47,wait for agv @18:         48,wait for agv @19:
       49,wait for agv @20:         50,wait for agv @21:
       51,wait for agv @22:         52,wait for agv @23:
       53,wait for agv @24:         54,wait for agv @25:
       55,wait for agv @26:         56,wait for agv @27:
       57,wait for agv @28:         58,wait for agv @28:
       59,type 1 flowtime:          60,type 2 flowtime:
       61,type 3 flowtime:          62,type 4 flowtime:
       63,type 5 flowtime:          64,type 6 flowtime:
       65,type 7 flowtime:          66,overall flowtime:
       67,wait for repair;
;
replicate,1,0,160;
;
end;
```

```
:----------------------------------------------------------------------:
:                                                                      :
:                          SIMAN Run Processor                         :
:                              Version 3.5                             :
:                        License Number 8810399                        :
:                                                                      :
:----------------------------------------------------------------------:
:                                                                      :
:   Systems Modeling Corporation licenses this program for use by :   :
:                                                                      :
:                         North Carolina State                         :
:                                                                      :
:   This program may only be used or copied according to the terms    :
:   of that license.                                                   :
:                                                                      :
:----------------------------------------------------------------------:


Please press <return> to begin the simulation.



          Recalling the   PROGRAM   file  level2.p


SIMAN Run Processor Version 3.5
Copyright 1985, 1986, 1987 by Systems Modeling Corp.




Beginning execution of run number    1
```

```
                          SIMAN Summary Report

                          Run Number    1 of  1

Project:  NADEP
Analyst:  WORSHAM
Date   :   7/12/1988

Run ended at time      .1600E+03


                            Tally Variables
                            ---------------

Number Identifier          Average     Standard    Minimum     Maximum     Number
                                       Deviation   Value       Value       of Obs
-------------------------------------------------------------------------------------

    1  WAIT FOR MACH  1      .00000      .00000      .00000      .00000          0
    2  WAIT FOR MACH  2      .00000      .00000      .00000      .00000          0
    3  WAIT FOR MACH  3      .00000      .00000      .00000      .00000          0
    4  WAIT FOR MACH  4      .42118     1.06193      .00000     4.55481         43
    5  WAIT FOR MACH  5      .68411     1.45702      .00000     4.40570         18
    6  WAIT FOR MACH  6      .08971      .26236      .00000     1.34547        107
    7  WAIT FOR MACH  7      .00000      00000       .00000      .00000          0
    8  WAIT FOR MACH  8      .00000      .00000      .00000      .00000          0
    9  WAIT FOR MACH  9      .00000      .00000      .00000      .00000         10
   10  WAIT FOR MACH 10      .00000      00000       .00000      .00000        110
   11  WAIT FOR MACH 11      .00000      .00000      .00000      .00000         22
   12  WAIT FOR MACH 12      .00000      .00000      .00000      .00000         30
   13  WAIT FOR MACH 13      .00000      .00000      .00000      .00000          0
   14  WAIT FOR MACH 14      .00000      .00000      .00000      00000           0
   15  WAIT FOR MACH 15     2.34506     3.47618      .00000    12.55348        109
   16  WAIT FOR MACH 16      00000       .00000      00000       .00000         45
   17  WAIT FOR MACH 17      .00000      .00000      00000       .00000          0
   18  WAIT FOR MACH 18     8.23333    12.20710      .00000    36.72751        105
   19  WAIT FOR MACH 19      00000       .00000      .00000      .00000          0
   20  WAIT FOR MACH 20     1 56715     2.78419      .00000     9.97118         26
   21  WAIT FOR MACH 21      .00000      .00000      .00000      .00000          0
   22  WAIT FOR MACH 22      .41365     1 26271      .00000     5.34975         18
   23  WAIT FOR MACH 23      .00000      .00000      .00000      .00000          0
   24  WAIT FOR MACH 24     2.85745     6 06007      .00000    19.66055         91
   25  WAIT FOR MACH 25      .00000      .00000      .00000      .00000          0
   26  WAIT FOR MACH 26      .00000      .00000      .00000      .00000          0
   27  WAIT FOR MACH 27      00000       .00000      00000       00000           0
   28  WAIT FOR MACH 28      .00000      .00000      .00000      .00000          0
   29  WAIT FOR MACH 29      .00000      .00000      .00000      .00000          0
   30  WAIT FOR AGV  01      .00000      .00000      .00000      .00000          0
   31  WAIT FOR AGV  02      .00000      .00000      .00000      .00000          0
   32  WAIT FOR AGV  03      .00000      .00000      .00000      .00000          0
   33  WAIT FOR AGV  04      .00152      .00300      .00000      .01300         42
   34  WAIT FOR AGV  05      00135       .00558      .00000      .02299         17
   35  WAIT FOR AGV  06      00166       .00394      .00000      .01300        107
   36  WAIT FOR AGV  07      00000       .00000      00000       .00000          0
   37  WAIT FOR AGV  08      .00000      .00000      .00000      .00000          0
   38  WAIT FOR AGV  09      00078       .00233      .00000      .00700          9
   39  WAIT FOR AGV 010      .00257      .00566      .00000      .02800        105
   40  WAIT FOR AGV 011      .00378      .00505      .00000      .01900         24
   41  WAIT FOR AGV 012      .00150      .00346      .00000      .01300         24
```

```
42  WAIT FOR AGV @13        .00000     .00000     .00000     .00000      0
43  WAIT FOR AGV @14        .00000     .00000     .00000     .00000      0
44  WAIT FOR AGV @15        .00291     .00608     .00000     .02300    107
45  WAIT FOR AGV @16        .00051     .00207     .00000     .01100     45
46  WAIT FOR AGV @17        .00000     .00000     .00000     .00000      0
47  WAIT FOR AGV @18        .00458     .00651     .00000     .02100    104
48  WAIT FOR AGV @19        .00000     .00000     .00000     .00000      0
49  WAIT FOR AGV @20        .00413     .00527     .00000     .01601     24
50  WAIT FOR AGV @21        .00000     .00000     .00000     .00000      0
51  WAIT FOR AGV @22        .00700     .00637     .00000     .01601     17
52  WAIT FOR AGV @23        .00000     .00000     .00000     .00000      0
53  WAIT FOR AGV @24        .00162     .00479     .00000     .02300     90
54  WAIT FOR AGV @25        .00000     .00000     .00000     .00000      0
55  WAIT FOR AGV @26        .00000     .00000     .00000     .00000      0
56  WAIT FOR AGV @27        .00000     .00000     .00000     .00000      0
57  WAIT FOR AGV @28        .00000     .00000     .00000     .00000      0
58  WAIT FOR AGV @28        .00000     .00000     .00000     .00000      0
59  TYPE 1 FLOWTIME        5.67777   19.98613     .00000   99.43565     64
60  TYPE 2 FLOWTIME       49.09554   24.90498   16.10904   72.02877      4
61  TYPE 3 FLOWTIME       41.08289    7.87390   29.38822   49.42799      6
62  TYPE 4 FLOWTIME       51.37981   23.34437   26.74033   77.65918      6
63  TYPE 5 FLOWTIME       34.58820   15.22874   15.26225   50.18332      6
64  TYPE 6 FLOWTIME       30.02503   11.19465   17.49320   39.03600      3
65  TYPE 7 FLOWTIME       31.53030   12.09067   18.78558   53.02036      5
66  OVERALL FLOWTIME      44.43307   21.28109   15.26225   99.43565     40
67  WAIT FOR REPAIR         .00000     .00000     .00000     .00000     50
```

Discrete Change Variables

Number	Identifier	Average	Standard Deviation	Minimum Value	Maximum Value	Time Period
1	# IN STN 1 QUE	.00000	.00000	.00000	.00000	160.00
2	# IN STN 2 QUE	.00000	.00000	.00000	.00000	160.00
3	# IN STN 3 QUE	.00000	.00000	.00000	.00000	160.00
4	# IN STN 4 QUE	.11319	.54010	.00000	5.00000	160.00
5	# IN STN 5 QUE	.07696	.29841	.00000	2.00000	160.00
6	# IN STN 6 QUE	.05999	.29160	.00000	3.00000	160.00
7	# IN STN 7 QUE	.00000	.00000	.00000	.00000	160.00
8	# IN STN 8 QUE	.00000	.00000	.00000	.00000	160.00
9	# IN STN 9 QUE	.00000	.00000	.00000	.00000	160.00
10	# IN STN 10 QUE	.00000	.00000	.00000	.00000	160.00
11	# IN STN 11 QUE	.00000	.00000	.00000	.00000	160.00
12	# IN STN 12 QUE	.00000	.00000	.00000	.00000	160.00
13	# IN STN 13 QUE	.00000	.00000	.00000	.00000	160.00
14	# IN STN 14 QUE	.00000	.00000	.00000	.00000	160.00
15	# IN STN 15 QUE	1.66510	2.63259	.00000	10.00000	160.00
16	# IN STN 16 QUE	.00000	.00000	.00000	.00000	160.00
17	# IN STN 17 QUE	.00000	.00000	.00000	.00000	160.00
18	# IN STN 18 QUE	5.40312	8.47054	.00000	29.00000	160.00
19	# IN STN 19 QUE	.00000	.00000	.00000	.00000	160.00
20	# IN STN 20 QUE	.25466	.67516	.00000	4.00000	160.00
21	# IN STN 21 QUE	.00000	.00000	.00000	.00000	160.00
22	# IN STN 22 QUE	.04654	.22762	.00000	2.00000	160.00
23	# IN STN 23 QUE	.00000	.00000	.00000	.00000	160.00
24	# IN STN 24 QUE	2.83531	6.58170	.00000	25.00000	160.00
25	# IN STN 25 QUE	.00000	.00000	.00000	.00000	160.00
26	# IN STN 26 QUE	.00000	.00000	.00000	.00000	160.00
27	# IN STN 27 QUE	.00000	.00000	.00000	.00000	160.00
28	# IN STN 28 QUE	.00000	.00000	.00000	.00000	160.00
29	# IN STN 29 QUE	.00000	.00000	.00000	.00000	160.00
30	UTIL OF STN 1	.00000	.00000	.00000	.00000	160.00
31	UTIL OF STN 2	.00000	.00000	.00000	.00000	160.00
32	UTIL OF STN 3	.00000	.00000	.00000	.00000	160.00
33	UTIL OF STN 4	1.71364	1.88315	.00000	8.00000	160.00
34	UTIL OF STN 5	.13906	.34603	.00000	1.00000	160.00
35	UTIL OF STN 6	.23861	.42624	.00000	1.00000	160.00
36	UTIL OF STN 7	.00000	.00000	.00000	.00000	160.00
37	UTIL OF STN 8	.00000	.00000	.00000	.00000	160.00
38	UTIL OF STN 9	.21346	.48667	.00000	2.00000	160.00
39	UTIL OF STN 10	2.96758	2.46350	.00000	10.00000	160.00
40	UTIL OF STN 11	.53742	.80163	.00000	4.00000	160.00
41	UTIL OF STN 12	.98007	1.19419	.00000	6.00000	160.00
42	UTIL OF STN 13	.00000	.00000	.00000	.00000	160.00
43	UTIL OF STN 14	.00000	.00000	.00000	.00000	160.00
44	UTIL OF STN 15	1.29038	.76472	.00000	2.00000	160.00
45	UTIL OF STN 16	.14802	.38084	.00000	2.00000	160.00
46	UTIL OF STN 17	.00000	.00000	.00000	.00000	160.00
47	UTIL OF STN 18	.16347	.36980	.00000	1.00000	160.00
48	UTIL OF STN 19	.00000	.00000	.00000	.00000	160.00
49	UTIL OF STN 20	1.03729	.80822	.00000	2.00000	160.00
50	UTIL OF STN 21	.00000	.00000	.00000	.00000	160.00
51	UTIL OF STN 22	.90998	1.20747	.00000	4.00000	160.00
52	UTIL OF STN 23	.00000	.00000	.00000	.00000	160.00
53	UTIL OF STN 24	.36238	.48069	.00000	1.00000	160.00

```
 54   UTIL OF STN 25          .00000      .00000      .00000      .00000     160.00
 55   UTIL OF STN 26          .00000      .00000      .00000      .00000     160.00
 56   UTIL OF STN 27          .00000      .00000      .00000      .00000     160.00
 57   UTIL OF STN 28          .00000      .00000      .00000      .00000     160.00
 58   UTIL OF STN 29          .00000      .00000      .00000      .00000     160.00
 59   # IN BEG AGV QUE        .00000      .00000      .00000      .00000     160.00
 60   # IN AGV QUE  @1        .00000      .00000      .00000      .00000     160.00
 61   # IN AGV QUE  @2        .00000      .00000      .00000      .00000     160.00
 62   # IN AGV QUE  @3        .00000      .00000      .00000      .00000     160.00
 63   # IN AGV QUE  @4        .00000      .00000      .00000      .00000     160.00
 64   # IN AGV QUE  @5        .00000      .00000      .00000      .00000     160.00
 65   # IN AGV QUE  @6        .00000      .00000      .00000      .00000     160.00
 66   # IN AGV QUE  @7        .00000      .00000      .00000      .00000     160.00
 67   # IN AGV QUE  @8        .00000      .00000      .00000      .00000     160.00
 68   # IN AGV QUE  @9        .00000      .00000      .00000      .00000     160.00
 69   # IN AGV QUE @10        .00000      .00000      .00000      .00000     160.00
 70   # IN AGV QUE @11        .00000      .00000      .00000      .00000     160.00
 71   # IN AGV QUE @12        .00000      .00000      .00000      .00000     160.00
 72   # IN AGV QUE @13        .00000      .00000      .00000      .00000     160.00
 73   # IN AGV QUE @14        .00000      .00000      .00000      .00000     160.00
 74   # IN AGV QUE @15        .00000      .00000      .00000      .00000     160.00
 75   # IN AGV QUE @16        .00000      .00000      .00000      .00000     160.00
 76   # IN AGV QUE @17        .00000      .00000      .00000      .00000     160.00
 77   # IN AGV QUE @18        .00000      .00000      .00000      .00000     160.00
 78   # IN AGV QUE @19        .00000      .00000      .00000      .00000     160.00
 79   # IN AGV QUE @20        .00000      .00000      .00000      .00000     160.00
 80   # IN AGV QUE @21        .00000      .00000      .00000      .00000     160.00
 81   # IN AGV QUE @22        .00000      .00000      .00000      .00000     160.00
 82   # IN AGV QUE @23        .00000      .00000      .00000      .00000     160.00
 83   # IN AGV QUE @24        .00000      .00000      .00000      .00000     160.00
 84   # IN AGV QUE @25        .00000      .00000      .00000      .00000     160.00
 85   # IN AGV QUE @26        .00000      .00000      .00000      .00000     160.00
 86   # IN AGV QUE @27        .00000      .00000      .00000      .00000     160.00
 87   # IN AGV QUE @28        .00000      .00000      .00000      .00000     160.00
 88   # IN AGV QUE @29        .00000      .00000      .00000      .00000     160.00
 89   # OF AGVS UP           9.65755      .54092     8.00000    10.00000     160.00
 90   # OF BUSY AGVS          .08728      .30121      .00000     3.00000     160.00
 91   # OF MACH @1 UP         .62876      .48314      .00000     1.00000     160.00
 92   # OF MACH @2 UP        1.00000      .00000     1.00000     1.00000     160.00
 93   # OF MACH @3 UP         .62321      .48458      .00000     1.00000     160.00
 94   # OF MACH @4 UP        7.28605      .75335     4.00000     8.00000     160.00
 95   # OF MACH @5 UP         .6[illegible]747      .37784      .00000     1.00000     160.00
 96   # OF MACH @6 UP        1.00000      .00000     1.00000     1.00000     160.00
 97   # OF MACH @7 UP         .96816      .17558      .00000     1.00000     160.00
 98   # OF MACH @8 UP         .86532      .34139      .00000     1.00000     160.00
 99   # OF MACH @9 UP        2.93069      .25399     2.00000     3.00000     160.00
100   # OF MACH @10 UP       9.91954      .30156     8.00000    10.00000     160.00
101   # OF MACH @11 UP       7.95056      .21677     7.00000     8.00000     160.00
102   # OF MACH @12 UP       7.80038      .44146     5.00000     8.00000     160.00
103   # OF MACH @13 UP       1.65661      .47484     1.00000     2.00000     160.00
104   # OF MACH @14 UP       1.95193      .21392     1.00000     2.00000     160.00
105   # OF MACH @15 UP       2.00000      .00000     2.00000     2.00000     160.00
106   # OF MACH @16 UP       5.00000      .00357     5.00000     5.00000     160.00
107   # OF MACH @17 UP       4.00000      .00000     4.00000     4.00000     160.00
108   # OF MACH @18 UP        .62866      .48316      .00000     1.00000     160.00
109   # OF MACH @19 UP        .90717      .29019      .00000     1.00000     160.00
110   # OF MACH @20 UP       1.96914      .17295     1.00000     2.00000     160.00
111   # OF MACH @21 UP        .96631      .18042      .00000     1.00000     160.00
112   # OF MACH @22 UP       3.59670      .49243     3.00000     4.00000     160.00
113   # OF MACH @23 UP       3.77792      .43791     2.00000     4.00000     160.00
```

```
 114   # OF MACH @24 UP        .87554       .33010      .00000     1.00000      160.00
 115   # OF MACH @25 UP       1.00000       .00000     1.00000     1.00000      160.00
 116   # OF MACH @26 UP       1.00000       .00000     1.00000     1.00000      160.00
 117   # OF MACH @27 UP        .77915       .41482      .00000     1.00000      160.00
 118   # OF MACH @28 UP        .72507       .44648      .00000     1.00000      160.00
 119   # OF MACH @29 UP       1.00000       .00000     1.00000     1.00000      160.00
 120   # IN REP QUEUE          .00000       .00000      .00000      .00000      160.00
 121   UTIL OF REPMEN         4.37566      1.91812      .00000    10.00000      160.00
 122   BATCHES IN SYS        21.35854     13.18775      .00000    43.00000      160.00
 123   BATCHES THRU SYS      15.93768     11.44835      .00000    40.00000      160.00
 124   BLADES THRU SYS         615.32       445.65         .00     1606.70      160.00

Run Time :  2 Minute(s) and 16 Second(s)


Stop - Program terminated.
```

Attachment 3

Project Data

NADEP Simulation Data Requirements Estimate

A) # Blade Repair Types = X = (20, 30, 40)		X	20			30			
B) Max # Repair Steps = 30									
C) Max # Machine Centers = Y = (20, 30, 40)		Y	20	30	40	20	30	40	
Data Associated with A		# Bytes							
Blade Names	(30 Char) * X	X * 30	600	600	600	900	900	900	
Yearly Quantity	(I*2) * X	X * 2	40	40	40	60	60	60	
Blades/Batch	(I*2) * X	X * 2	40	40	40	60	60	60	
Frequency Dist.	(2 * R*4 & 2 Char) * X	X * 10	200	200	200	300	300	300	
Priority	(R*4) * X	X * 4	80	80	80	120	120	120	
Data Associated with B									
Repair Steps	(I*2) * X * 30	X * 2 * 30	1200	1200	1200	1800	1800	1800	
Machine Times	(2 * R*4 & 2 Char) * X * 30	X * 10 * 30	6000	6000	6000	9000	9000	9000	1.
Data Associated with C									
Machine #	(I*2) * Y	Y * 2	40	60	80	40	60	80	
Machine Nomenclature	(30 Char) * Y	Y * 30	600	900	1200	600	900	1200	
# Each Type Mach.	(I*2) * Y	Y * 2	40	60	80	40	60	80	
Reliability Dist.	(2 * R*4 & 2 Char) * Y	Y * 10	200	300	400	200	300	400	
Queue Size	(I*2) * Y	Y * 2	40	60	80	40	60	80	
Setup	(2 * R*4 & 2 Char) * X * Y	X * Y * 10	4000	6000	8000	6000	9000	12000	
Teardown	(2 * R*4 & 2 Char) * X * Y	X * Y * 10	4000	6000	8000	6000	9000	12000	
Machine States	(I*2) * Y	Y * 2	40	60	80	40	60	80	
Mach. Repair Times	(2 * R*4 & 2 Char) * Y	Y * 10	200	300	400	200	300	400	
Transportation System									
Travel Time Matrix	(2 * R*4 & 2 Char) * Y^2	Y^2 * 10	4000	9000	16000	4000	9000	16000	
Totals			21,320	30,900	42,480	29,400	40,980	54,560	37

# Blade Types	# Machining Centers		
	20	30	40
20	21,320	30,900	42,480
30	29,400	40,980	54,560
40	37,480	51,060	66,640

NADEP Simulation Data Requirements
Estimate Summary
(Numbers Represent Bytes)

NADEP Engine Blade Rework Facility Simulation Data

Blades Data

Blade Names	Yearly Quantity	Blades/ Batch	Priorities
TF30 3rd Stage Turbine Blade	10410	92	1.00
F402 Z-Notch Repair	7738	100	1.00
F402 Snubber Repair	7738	50	1.00
TF30 4th Stage Turbine Blade	12000	90	1.00
TF30 2nd Stage Turbine Blade	9200	94	1.00
J 52 Class 13 1st Stage Vane AS	10000	100	1.00
F402 2nd Stage HPT 2 Vane	8200	50	1.00

Calendar Data

Average Number of Hours in a Fiscal Year:	1920
Number of Shifts Worked per Day:	2

Model Specifications

Number of Blade Types	7
Number of Machine Centers	28
Maximum Number of Queued Jobs for any Shop	40

Machine Data

Stn #	Mach ID	Machine Nomenclature	Old Capacity	New Capacity
1	212	Sputter Coater	1	3
2	900	Mini-Hipper	1	N/A
3	207	Production-Hipper	1	1
4	211	Dabber Welder	8	4
5	214	Low Pressure Plasma Spray	1	1
6	201	Low Pressure Penetrant Ins	1	1
7	203	Auto Elec. Optical Scanner	1	1
8	210	Braze Cleaning	1	1
9	236	Plasma Needle Arc Welder	2	2
10	235	Grinding & Polishing Machine	10	10
11	221	Airfoil Grinder	2	2
12	935	Electrolytic Surface Grinder	7	N/A
13	234	Spot Welder	2	2
14	219	Laser Marker	2	2
15	950	Cleaning Line	2	N/A
16	205	Abar Furnace	5	6
17	960	Ultra Sonic Cleaner	4	0
18	340	Degreaser	1	1
19	228	Aluminum Oxide Blast	1	1
20	213	5-Axis Machine	2	4
21	980	Small Furnace	1	N/A
22	222	Drill EDM	2	2
23	223	Wire EDM	2	2
24	227	Glass Bead Blast	1	1
25	991	Vibratory Finisher	1	N/A
26	992	De-Ionized Water Tank	1	N/A
27	993	Allison Electrophoratic Proc	1	1
28	994	Outside Router	1	N/A

* When new capacities were not available, old capacities assumed

Blade Routes

Blade #	1	2	3	4	5	6	7
# Stations Visited	14	14	14	13	13	18	8
Interarrival Time Avg.	34	0	25	29	78	38	23
1st Station Visited	15	18	15	6	6	18	18
Service Time	0.50	0.50	0.50	0.27	0.27	0.50	0.25
2nd Station Visited	6	15	15	10	10	24	22
Service Time	0.27	2.00	1.50	4.00	4.00	1.50	8.00
3rd Station Visited	10	24	15	24	24	15	10
Service Time	4.00	1.00	2.50	0.50	0.50	2.50	6.25
4th Station Visited	24	6	6	18	18	15	15
Service Time	0.50	1.00	1.00	0.21	0.21	1.00	5.00
5th Station Visited	18	16	20	4	4	18	24
Service Time	0.21	1.00	8.00	8.50	8.50	0.50	1.00
6th Station Visited	4	24	15	12	12	10	9
Service Time	8.50	1.50	1.50	8.00	8.00	8.00	4.00
7th Station Visited	12	18	15	11	11	18	5
Service Time	8.00	0.50	5.00	5.00	5.00	0.50	4.00
8th Station Visited	11	4	4	10	10	15	5
Service Time	5.00	8.00	8.00	8.00	8.00	2.50	1.50
9th Station Visited	10	20	16	6	6	24	
Service Time	8.00	8.00	1.00	0.27	0.27	1.50	
10th Station Visited	6	12	20	18	18	16	
Service Time	0.27	8.00	16.00	0.21	0.21	1.50	
11th Station Visited	18	10	10	16	16	24	
Service Time	0.21	10.00	10.00	0.50	0.50	1.50	
12th Station Visited	16	6	10	24	24	15	
Service Time	0.50	1.00	2.50	0.50	0.50	2.50	
13tn Station Visited	24	10	6	6	6	4	
Service Time	0.50	1.50	1.00	0.27	0.27	10.00	
14th Station Visited	6	10	24			16	
Service Time	0.27	2.50	1.50			1.50	

Blade #	1	2	3	4	5	6	7
15th Station Visited						6	
Service Time						1.00	
16th Station Visited						10	
Service Time						16.00	
17th Station Visited						6	
Service Time						1.00	
18th Station Visited						24	
Service Time						1.50	
Tot Service Time/Batch	36.73	46.50	60.00	36.23	36.23	55.00	30.00

NADEP Blade & Vane Repair Facility

Distances Between Stations

	To															
From	4	5	6	9	10	11	12	15	16	18	20	22	23	24	30	31
4	--	11	28	7	20	13	8	32	4	31	16	32	20	19	31	32
5		--	20	16	30	23	18	30	8	25	26	24	30	11	23	24
6			--	32	19	26	31	13	25	8	23	7	19	12	3	4
9				--	14	7	2	26	10	25	10	26	14	25	35	36
10					--	8	13	13	23	12	5	13	2	28	22	23
11						--	6	20	16	19	4	20	8	31	29	30
12							--	25	11	24	9	25	13	26	34	35
15								--	36	7	18	8	13	23	16	17
16									--	30	19	29	23	16	28	29
18										--	16	2	12	17	11	12
20											--	17	5	32	26	27
22												--	13	16	10	11
23													--	28	22	23
24														--	15	16
30															--	1
31																--

* Distances approximate yards = 1 grid on Cinema layout.

Bibliography

Bkehrens, Matthew K., "Star-Dot-Star, All Keyed Up", *PC World*. Volume 6, No. 10, October 1988, pp. 241-242.

CINEMA, System for Simulation and Animation, Systems Modeling Corp., State College, PA, 1988.

Introduction to Simulation with SIMAN. Systems Modeling Corp., State College, PA, 1987.

Jabarin, Dorothy H. and Arleen G. Schwartz, *The Book of BASIC*, COMPUTE! Publications, Inc., Greensboro, NC, 1984.

McMullan, William, "Menus a la Carte", *80 Micro*, IDG Communications, March 1988, pp. 36-38.

Pegden, Dennis C., *Introduction to SIMAN with Version 3.0 Enhancements*, Systems Modeling Corp., State College, PA, 1986.

Tandy 1000 BASIC Reference Manual, Tandy Corporation, Fort Worth, TX, 1986.

Tandy 1000 MS-DOS Reference Manual, Tandy Corporation, Fort Worth, TX, 1986.

The SIMAN Simulation Language Reference Guide, SIMAN Simulation Software, Systems Modeling Corp., State College, PA, 1987.